N I

A COMPUTER PROGRAM FOR FUEL SLOSHING IN AN AXISYMMETRIC TANK

by

W. H. Chu
R. Gonzales
A. F. Muller
D. R. Saathoff

TM Harry Buchanan

TECHNICAL REPORT NO. 10

Contract No. NAS8-20290
Control No. DCN 1-9-75-10050(1F)
SwRI Project No. 02-1846-02

Prepared for

National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812
George C. Marshall Space Flight Center

August 1970

SOUTHWEST RESEARCH INSTITUTE
SAN ANTONIO HOUSTON

SOUTHWEST RESEARCH INSTITUTE
Post Office Drawer 28510, 8500 Culebra Road
San Antonio, Texas 78228

A COMPUTER PROGRAM FOR FUEL SLOSHING IN AN AXISYMMETRIC TANK

by,

W. H. Chu
R. Gonzales
A. F. Muller
D. R. Saathoff

TECHNICAL REPORT NO. 10

Contract No. NAS8-20290
Control No. DCN 1-9-75-10050(1F)
SwRI Project No. 02-1846-02

Prepared for

National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812
George C. Marshall Space Flight Center

August 1970

Approved:

H. Norman Abramson, Director
Department of Mechanical Sciences

TABLE OF CONTENTS

LIST OF TABLES

I. DESCRIPTION OF MAIN PROGRAM

A. Program Name

The name of the main program is TRIPOT, meaning a potential problem solved by a non-uniform triangular mesh.

B. Functions of TRIPOT

1. Generation of a Non-Uniform Triangular Mesh

The non-uniform triangular mesh is generated on computer by the Winslow method (Ref. 1). The essence of the method is to use a transform to a logical plane, ξ, η which is related to the physical plane x, y by

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0, \quad \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0 \tag{1a, b}$$

and by interchanging dependent and independent variables

$$\alpha x_{\eta\eta} + \beta x_{\xi\eta} + \gamma x_{\xi\xi} = 0 \tag{2a}$$

$$\alpha y_{\eta\eta} + \beta y_{\xi\eta} + \gamma y_{\xi\xi} = 0 \tag{2b}$$

where

$$\alpha = x_\xi^2 + y_\xi^2 \tag{3a}$$

$$\beta = x_\eta x_\xi + y_\eta y_\xi \tag{3b}$$

$$\gamma = x_\eta^2 + y_\xi^2 \tag{3c}$$

On the logical plane ξ, η; the domain is composed of equilateral triangles only.

Equations (2a, b) are solved by specifying boundary values of x, y on the selected logical domain ξ, η. For fuel sloshing in an axisymmetric tank we chose a simple parallelogram as our logical domain (Fig. 3, Appendix II).* The interface of arc length s_1 is located on $i = 1$, $j = 1$ to N. The centerline of length s_2 is between the interface and the tank bottom. The wall is divided into two parts; the "bottom" of arc length s_3 and the "side" of arc length s_4. For cylindrical tanks, the side and bottom are distinct; hence it will be used as such and not be redefined, and then s_2 and s_4 are divided into M parts, while s_3 is divided into N parts as s_1. For a tank such as a spherical tank, the wall is arbitrarily divided into s_3 and s_4 approximately in the ratio s_1 to s_2. Parts of approximately equal length can be specified. For fuel sloshing, especially with a "folding" interface, more accuracy near the contact point and the folding part may be desirable (as the machine time and the storage locations increase rapidly with number of points on the interface and the side wall). Then one may use a coarser net near the center on the interface and away from the contact point on the side wall, while s_2 and s_3 are divided in parallel with s_4 and s_1, respectively. One may also use a finer net near the junction of s_3 and s_4. In general, it is preferable to generate nearly equilateral triangles and to avoid obtuse triangles (not necessarily eliminated in the Winslow method). For a folding interface, relatively high accuracy of the interface points is needed, and can be fed in from outputs of the interface program, SSHAPE, meaning surface

*All figures in the theory are given in Appendix II.

shape (see Appendix I). For example, one may use N = 17 in SSHAPE (there are 16 intervals but take 5 double intervals near the center and 6 single intervals elsewhere), then in TRIPOT use N = 12 (there are 11 intervals). Likewise, a net of N = 24 may be obtained with piecewise uniform spacing.

With boundary values of x, y specified, the interior net points are first solved by a "linearized" approximation (Ref. 1), namely

$$\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial^2 x}{\partial \eta^2} = 0, \quad \frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 y}{\partial \eta^2} = 0 \tag{4a, b}$$

Then the technique of combined underrelaxation of coefficients and overrelaxation of dependent variables is used to solve Eq. (2a, b) starting with solutions of Eq. (4a, b). For Eq. (4a, b), an initial relaxation factor of WA = 1.8 is used. For Eqs. (2a, b), an initial overrelaxation factor of WB = 1.0 is used. The overrelaxation factors are improved as given in reference 1. For example,

$$\eta^n_{\text{Winslow}} = \text{ETAX} = \left[\frac{\sum_i\sum_j (x^{n+1}_{ij} - x^n_{ij})^2}{\sum_i\sum_j (x^n_{ij} - x^{n-1}_{ij})^2}\right]^{1/2} \tag{5}$$

$$\lambda_{\text{Winslow}} = \text{ELX} = \left[\frac{\omega^n + \eta^n - 1}{\omega^n\sqrt{\eta^n}}\right]_{\text{Winslow}}$$

$$= \frac{\text{WX} + \text{ETAX} - 1}{\text{WX} * \sqrt{\text{ETAX}}} \tag{6}^\dagger$$

†WX is the overrelaxation factor.

$$(\omega'_{opt})_{Winslow} = WAX = \left[\frac{2}{1 + (1 - \lambda^2)^{1/2}} - \omega_0\right]_{Winslow}$$

$$= \frac{2}{1 + \sqrt{1 - ELX{*}{*}2}} - W0; \quad W0 \cong 0.01 \qquad (7)$$

$$(\omega^{n+1})_{Winslow} = WX = [\beta\omega'_{opt} + (1 - \beta)\,\omega^n]_{Winslow}$$

$$= RHO * WAX + (1 - RHO)\,WX; \quad RHO = 0.05 \qquad (8)$$

It is noted that if ETAX > 1, use WAX = WX (previous) instead of Eq. (7).

Let

$$XTEMP = [C_1 x^n_{i-1,j} + C_2 x^n_{i-1,j-1} + C_3 x^n_{i,j-1} + C_4 x^n_{i+1,j} + C_5 x^n_{i+1,j+1} + C_6 x^n_{i,j+1}]/SUMC \qquad (9)$$

$$SUMC = \sum_{k=1}^{6} C_k \qquad (10)$$

$$RX = [x^n_{i,j} - XTEMP] * WX \qquad (11)$$

then

$$x^{n+1}_{i,j} = x^n_{i,j} - RX \qquad (12)$$

It is noted that in the present problem, we use a simple parallogram in the logical plane (Fig. 3, Appendix II). Two examples of constructed nets in a physical plane are shown in Figures 4a and 4b, Appendix II.

2. Sequence of Calculations

The theory using auxiliary characteristic functions ψ_j is given in Appendix II. They satisfy $\frac{\partial\psi_j}{\partial n} = \lambda\psi_j$ on F, $\frac{\partial\psi_j}{\partial n} = 0$ on W and $\nabla^2\psi = 0$ in V_L.

In our program, ψ_j is found by the influence coefficient method. The influence coefficients are F_{jk} and, therefore, one has

$$\psi_j = F_{jk} \frac{\partial \psi_k}{\partial n} = \lambda F_{jk} \psi_k \text{ on } F \tag{13}$$

or

$$\left([F] - \frac{1}{\lambda}[I]\right)\left\{\psi\right\} = 0 \text{ on } F \tag{14}$$

which is a standard eigenvalue problem; the k^{th} eigenvector being ψ_k, and the corresponding λ_k are printed as eigenvalues. The k^{th} column of F_{jk} is constructed with a unit k^{th} element of $\frac{\partial \psi}{\partial n}$ and zero normal derivative elsewhere. The value of k varies from 2 to N while $\psi(1, 1) = 0$ and $\frac{\partial \psi(1,1)}{\partial n} = 0$, at the center of F. ψ is governed by

$$\nabla^2 \psi = 0 \tag{15}$$

A finite difference formula of Eq. (15) on the triangular mesh was given in reference 1. Let

A_{ij} = the area of ij^{th} dodecagon inside the fluid domain

r_{ij} = radius of the ij^{th} point

$w_k = \frac{1}{2}(\lambda_k \overline{r}_k \cot \theta_k + \lambda_{k-1} \overline{r}_{k-1} \cot \sigma_k)$, k = 1 to 6 (see Fig. 1a, b)†

$\overline{r}_k = \frac{1}{3}(r_{ij} + r_k + r_{k+1})$

† θ_k, σ_k can be expressed in terms of t_k, u_k, t_{k-1}, u_{k-1} and s_k.

For an interior point (i, j),

$$\sum_{k=1}^{6} w_k(\psi_k - \psi) - \frac{m^2}{r_{ij}} A_{ij}\psi = 0, \quad m = 1 \tag{16}$$

with

$$\lambda_k = 1$$

For an interface point (1, j),

$$\sum_{k=1}^{6} w_k(\psi_k - \psi) - \frac{m^2}{r_{1j}} A_{ij}\psi + \left(\frac{\partial\psi}{\partial n}\right)_{1j} [\tfrac{1}{2} s_3 + \tfrac{1}{2} s_6] \; r_{1j} = 0 \tag{17}$$

with

$$\lambda_6 = \lambda_1 = \lambda_2 = 0, \; \lambda_3 = \lambda_4 = \lambda_5 = 1, \; i = 1$$

For ij^{th} point on the tank wall,

$$\sum_{k=1}^{6} w_k(\psi_k - \psi) - \frac{m^2}{r_{ij}} A_{ij}\psi = 0, \quad m = 1 \tag{18}$$

with

$\lambda_3 = \lambda_4 = \lambda_5 = 0$ and $\lambda_1 = \lambda_2 = \lambda_6 = 1$ on bottom wall,

i.e., $i = M$, $j = 1$ to N

$\lambda_4 = \lambda_5 = \lambda_6 = 0$ and $\lambda_1 = \lambda_2 = \lambda_3 = 1$ on side wall, i.e.,

$j = N$, $i = 1$ to M

$\lambda_1 = \lambda_2 = 1$ and $\lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = 0$ at corner point,

$i = M$, $j = N$

On centerline,

$$\psi = 0 \text{ for } m = 1 \tag{19}$$

At contact point, $i = 1$, $j = N$

$$\sum_{k=1}^{6} w_k(\psi_k - \psi) - \frac{m^2}{r_{ij}} A_{ij}\psi + \left(\frac{\partial\psi}{\partial n}\right)_{1,N} \left(\frac{1}{2} s_3\right) r_{ij} = 0 \tag{20}$$

with

$$\lambda_3 = 1 \text{ and } \lambda_4 = \lambda_1 = \lambda_2 = \lambda_5 = \lambda_6 = 0$$

Equations (16) to (20) are solved by the overrelaxation method through use of subroutine SOREL (1), which contains the same procedure and factors as those used in solving for x [see Eqs. (8), (12)].

After the influence coefficient matrix is determined, Eq. (14) determines the eigenvalues and eigenvectors (the auxiliary characteristic functions ψ_j on the interface). The values of ψ_j on the wall are obtained by the S. O. R. method through subroutine SOREL (2).

Then subroutine SOREL (1) is again used to solve the boundary value problem of a capped fluid (rigid interface) under pitching oscillation and subsequently the corresponding moment of inertia.

Next, the normal modes are determined by an expansion of auxiliary characteristic functions (using N_{ev} numbers† out of the N number

†It is probably more accurate not to use the last few auxiliary eigenfunctions which contain larger errors.

of $\psi_j^{\prime}$s). A matrix eigenvalue problem results as given in Appendix II. The computer program thus calculates various integrals entering the matrix equation. An eigenvalue subroutine yields the coefficients in series expansion of the normal modes (eigenvectors) and the corresponding frequency (eigenvalue).

Once the coefficients in the series expansion of the normal modes are determined, integrals containing these functions (Appendix II) can be evaluated to determine the parameters of the spring-mass mechanical model. This completes the main steps of the present program.

II. EXAMPLES

Results of numerical examples are given in Appendix III.

It is noted that for a spherical tank with a "folding" interface, the result is sensitive to the inputs, especially the interface points; piecewise uniform spacings of the interface points were used. With a 12×12 mesh, it takes about two minutes CDC 6600 time in the main program, TRIPOT. With 23×34 mesh, it would take about twenty minutes.

III. CONCLUSIONS

The program seems to yield reasonably good results in the examples calculated with as few as 12 interface points. However, in some cases, it is sensitive to inputs which should be reasonably accurate. Applications to other cases remain to be examined.

IV. INSTRUCTIONS FOR USING COMPUTER PROGRAMS "TRIPOT" AND "SSHAPE"

A. General

TRIPOT is a program that computes the sloshing parameters for an axisymmetric tank. A finite-difference technique using a triangular mesh is used to solve the potential flow equations. The main restriction on the program is that the Bond number and the volume of contained liquid are both great enough so that the liquid covers the bottom of the tank.

SSHAPE is a program used to compute the equilibrium free surface shape of the liquid. The output of SSHAPE is used as input for TRIPOT.

B. TRIPOT

A triangular mesh is laid over the liquid by a method to be described. Figures 4a and 4b, in Appendix III, show typical meshes for a cylindrical and a spheroidal tank.

Both the liquid interface and the tank shape are given as discrete x, y coordinates, $x = 0$, $y = 0$ being the point where the free surface intersects the tank centerline. The interface, arc length s_1, is broken up into N points. The centerline, arc length s_2, is broken up into M points. The bottom, s_3, and side wall, s_4, are specified at N and M points, respectively. For a spheroid, for example, in which there is no "bottom" or "side," the wall is arbitrarily divided into arc lengths s_3 and s_4 such that s_3 / s_4 is approximately equal to s_1 / s_2.

For a "folded over" interface, additional accuracy can be obtained by using a non-uniform spacing of the N and M points. A coarse net is used on the interface, s_1, near the centerline and on the part of the wall not close

to the contact point, s_4. Then, s_2 and s_3 are divided similarly to s_1 and s_4, respectively. For example, 16 intervals (N = 17) can be used in SSHAPE to calculate the free surface shape. Then, the s, y coordinates of the 1, 3, 5, 7, 9, 11, 12, 13, 14, 15, 16, 17 points can be used as input to TRIPOT with N = 12 (eleven intervals, 5 wide ones near the centerline and 6 narrow ones near the contact point).

The interior grid points are generated by TRIPOT. In general, it is better to use near-equilateral triangles and avoid obtuse ones. Some trial-and-error work may be necessary to insure this.

Subprograms Used

SOREL - Solution of a system of equations by successive over-relaxation

SFIT - Curve fitting by a second degree polynomial

EIGEN - Eigenvalues and vectors of a real symmetric or non-symmetric matrix

MATINV - Matrix inversion routine

Tapes Used

TAPE 6 - Restart write tape

TAPE 7 - Restart read tape

Storage Used

Less than 32,000 case storage locations

Input Formats Used

Card No.	Format
1	(12A6)
2	(4I5, 5F10.0)
3†	(8F10.0)
4†	(8F10.0)
5†	(8F10.0)
6†	(8F10.0)
7	(8F10.0)
8	(5E15.8)
9	(4I5, 5F10.0)

C. TRIPOT--Program Restart Procedure

At seven different places in the program, results calculated thus far are saved on tape so that it is not necessary to calculate from the beginning in the event of catastrophic failure or insufficient alloted time. When using the restart feature it is necessary to interchange tape six with tape seven so that the tape which was written will become the tape being read. One method of effecting the interchange is by redefining LUN and KUN, which are the first two executable statements in the main program. During execution, the printed output will state the highest restart number which can be used. The input data card numbers to be used with each restart are given below.

Restart Number	Input Data Card Numbers
1	1, 2, 7, 8, 9
2	1, 2, 7, 8, 9
3	1, 2, 7, 8, 9
4	1, 2, 8, 9
5	1, 2, 8, 9
6	1, 2, 8, 9
7	1, 2, 8, 9

†Set of cards.

TABLE I. INPUT OF TRIPOT

TRIPOT--Input Data Description

Card No.	FORTRAN Symbol	Variable Name	Units	Definition
1	ITITLE			72 columns of identification
2	M			Number of centerline points
	N			Number of interface points
	ITA			Maximum number of iterations
	IRS			Restart number (initially zero)
	WA		non-dim.	Relaxation factor for linear approximations
	WB		non-dim.	Initial relaxation factor for nonlinear solution
	EPS		non-dim.	Convergence factor
	RLGTH		length	Reference length used for non-dimensionalization, e.g., maximum tank radius
	BOND	N_B	non-dim.	Bond number
3†	X(1, J), Y(1, J)	(x, y)j=1, N	length	Coordinates for interface‡; j = 1 at tank center, increasing outward
4†	X(I, N), Y(I, N)	(x, y)i=2, M-1	length	Coordinates for wall; i = 1 at interface increasing downward
5†	X(M, J), Y(M, J)	(x, y)j=1, N	length	Coordinates for centerline, j = 1 at tank center, increasing outward
6†	X(I, 1), Y(I, 1)	(x, y)i=2, M-1	length	Coordinates for centerline, i = 1 at interface increasing downward

†Set of cards.

‡It is recommended to use 5 digits (or more) obtained from output of interface program. Wall coordinates are not as critical.

TABLE I (cont'd)

Card No.	FORTRAN Symbol	Variable Name	Units	Definition
7	GFLC		non-dim.	= 0, compute Γ ; input dummy Γ = 1, input Γ from SSHAPE output, or if interface is flat input $\Gamma = 0$.
	GAMMA	Γ	non-dim.	Contact point constant
	THETAC	θ_c	degrees	Contact angle; must be greater than 2°
8	ZCG		length	Center of gravity of liquid†
	VOL		length cube	Volume of liquid†
	DEN		mass/length cube	Density of liquid
	RHOU		mass/length cube	Density of ullage fluid
9	NEV	N_{ev}	non-dim.	Number of eigenvectors used

†Obtained as output from SSHAPE.

D. TRIPOT Input Example and General Instructions

Example for a Spheroidal Tank and General Input† Instructions are given below:

1. TEST CASE xx, x Full, Bond number = x (see input description)

2. a. M = 9 — no. of side wall points, see 5 below

   b. N = 12 — no. of interface points selected, see 4 below; usually greater than 10

   c. ITA = 200 — suggested value; usually sufficient

   d. IRS = 0 — always zero unless using restart procedure

   e. WA = 1.8 — suggested value

   f. WB = 1.3 — suggested value

   g. EPS = 1×10^{-6} — suggested value

   h. RLGTH = 1.0 — (see input data description)

   i. Bond = 5 — specified value

3. X(1, J), Y(1, J) — selected interface points from output of SSHAPE

4. X(I, N), Y(I, N) — side wall points selected after interface is calculated; a plot of interface, tank wall and center is desirable for selecting wall points. Suggest division near contact point be nearly an isocicle or form an acute triangular. It is not necessary to have precisely equal arc divisions. Usually, the meshes may be finer near the contact point. The points can be read off from curve (or calculated from analytic expression); accuracy is not as critical as the interface points.

†TRIPOT input data descriptions are given in Table I.

5.	X(M, J), Y(M, J)	bottom wall points; the ratio of bottom wall to side wall in arc length is approximately that of interface to wetted centerline (eye observation is usually sufficient). The divisions on the bottom are suggested to be approximately parallel to that on the interface. Points can usually be read off from curve plotted.
6.	X(I, 1), Y(I, 1)	wetted centerline points. The divisions on the centerline is suggested to be approximately parallel to that on the side wall.
7.	a. GFLC = 1.0	(see input data description) unity using output from SSHAPE
	b. GAMMA = -16.12	output CGAM from SSHAPE
	c. THETAC = 5°	specified contact angle (no less than 2°)
8.	a. ZCG = -.01	output from SSHAPE; c.g. of liquid, negative below center of interface
	b. VOL = 1.35	output from SSHAPE; liquid volume
	c. DEN = 1.0	if RHOU = 0, the density of the liquid DEN can be set to 1 without affecting result
	d. RHOU = 0	density of ullage fluid usually neglected (zero)
9.	NEV = 7	number of eigenvectors used; for $N \geq 11$, $N_{ev} = 7$ has been selected. For partial convergence test on a finer mesh $N_{ev} = 7$ can be retained and for the convergence test on Galerkin procedure, N_{ev} may be increased. However, it may be advisable to use $N_{ev} < N-3$ or smaller to avoid large inaccuracy in the higher auxiliary characteristic functions.

E. <u>TRIPOT Output</u>

TRIPOT--Program Printed Output

1. Input Data - ITITLE, M, N, ITA, WA, WB, EPS, IRS
2. (x, y) coordinates for linear approximations to the mesh

3. (x, y) coordinates for nonlinear solution to the mesh

4. Input Data (compute option) - GAMMA

5. Eigenvalues λ_j and eigenvectors ψ_{m_j} on F

6. Input Data - BOND, ZCG, VOL, DEN, RHOU

7. Eigenvalues $\Omega_k^2 = \dfrac{\omega_k^2 a}{g}$ and eigenvectors C_{k_j}

8. SIF, $I_F / M_F h_o^2$

9. SIO, $I_o / M_F h_o^2$

10. MK/MF_1 - nondimensional i^{th} slosh mass, M_K/M_F

11. ZK_i, k^{th} slosh mass location, z_k with $i = k = 1, 2 \ldots NEV$

12. Z0, location of rigid mass, z_0

REFERENCES OF THE MAIN COMPUTER PROGRAM

1. Winslow, A. M., "Numerical Solutions of the Quasi-Linear Poisson Equations in a Non-Uniform Triangular Mesh," Journal of Computational Physics, Vol 1, No. 2, pp 149-172, November 1966.

2. See references in Appendix II.

APPENDIX I

A COMPUTER PROGRAM FOR GENERATING THE INTERFACE SHAPE AND ASSOCIATED QUANTITIES

A. Governing (Mean) Interface Equation

The interface differential equation is given by†

$$y'' = \frac{2y^2 + 3(y')^2}{y} - \frac{y'}{y^2} \cot(\theta) [y^2 + (y')^2] + \frac{1}{y} \left[B_n (y \cos(\theta) - y_0) - \frac{2k_0}{y_0} \right] [y^2 + (y')^2]^{3/2} \tag{I-1}$$

Converting from second order to first order,

let

$$y_1 = y$$
$$y_2 = y'$$

then

$$y_1' = y'$$
$$y_2' = y'' \tag{I-2}$$

or

$$y_1' = y_2$$

$$y_2' = \frac{2y_1^2 + 3y_2^2}{y_1} - \frac{y_2}{y_1^2} \cot(\theta) [y_1^2 + y_2^2] + \frac{1}{y_1} \left\{ B_n [y_1 \cos(\theta) - y_0] - \frac{2k_0}{y_0} \right\} (y_1^2 + y_2^2)^{3/2} \tag{I-3}$$

†Hastings, L. J. and Rutherford, R., III, "Low Gravity Liquid-Vapor Interface Shapes in Axisymmetric Containers and a Computer Solution," NASA Technical Memorandum, NASA TM X-53790, October 7, 1968. In the equation to follow, y is the radial distance from the center of the top of the tank, θ is the angle counterclockwise from the tank centerline with origin at the center of the top.

The solution of the first order differential equations (I-2) and (I-3) is obtained by using the Runge-Kutta-Gill fourth order method; the boundary of the tank is approximated by a polygon. In general, a new polygonal approximation is required below the interface according to the description aforementioned in the main text.

B. Program Notes

In addition to the main program, the following subprograms were used:

RKLDEQ - Runge-Kutta-Gill differential equation solver

SFIT - curve fitting by a 2nd degree polynomial.

The input data description of program SSHAPE is given in Table II and the output of program SSHAPE is given in Table III, respectively. Some of the program symbols which were not given previously are defined in Table IV and an example keypunch form is given in Table V.

C. Input Instructions for SSHAPE

The input for a spherical tank and general instructions are given below:

1.	NP = 37	Total number of points specifying the tank starting from center of bottom to center of top; varies with each case
2.	YBX_i	Horizontal and vertical coordinates of tank; should be a close polygonal approximation to the tank, i = 1 starting from center of tank bottom
3.	YBZ_i	
4.	a. $\alpha = 5°$	Contact angle specified
	b. KO = 0	Usually zero

	c. YO = 1.0	Initial guess at distance between center of interface and center of tank top
	d. BN = 5	Bond number specified
	e. BETAD = 5/8 = .625	Empty fraction specified
	f. DKO = 0.2	Suggested
	g. DBC = .005	Convergent criterion
	h. DYO = .05	Suggested
5.	a. THETA = 0	Usually zero
	b. DTHETA = .05	Suggested
	c. RLGTH = 1	(See input data description)
6.	a. N = 2	Always
	b. IBOPT = 1	Unity if calculating empty fraction (see input data description)
7.	a. NN(1) = 17	(Only 12 of these 17 were selected for input in TRIPOT)
	b. NN(2) = 33	For a finer mesh if desired

D. <u>Listing and Sample Input</u>

These are included after those of TRIPOT at the end of the report.

TABLE II

INPUT DATA DESCRIPTION OF PROGRAM SSHAPE

Card No.	FORTRAN Symbol	Variable Name	Units	Definition
1	NP			No. of points on container boundary
2	YBX_i		length	x-coordinate container boundary, positive outward from top center, i = 1 at bottom center
3	YBZ_i		length	z-coordinate container boundary, positive downward from top center, i = 1 at bottom center
4	ALPHA	α	degrees	Desired contact angle, θ_c , must be greater than 2°
	KO	K_o	non-dim.	Assumed initial guess of parameter related to curvature k_o at inter-face center point, i.e. $k_o y_o$. Usually zero can be used.
	YO	y_o	length	Estimated distance from center of tank top to interface centerpoint
	BN	B_n	non-dim.	Bond number, N_{B_e}
	BETAD	β_D	non-dim.	Desired empty fraction
	DKO	ΔK_o	non-dim.	Increment for K_o , a typical value is 0.2.
	DBC	$\epsilon_{\beta D}$	non-dim.	Convergence criterion for β_D. (A typical value is 0.005)
	DYO	Δy_o	non-dim.	Increment for y_o . (A typical value is 0.05 R_o).
5	THETA	θ	degrees	Initial angle measured from vertical axis to y . Usually zero.

TABLE II

INPUT DATA DESCRIPTION OF PROGRAM SSHAPE (Cont'd)

Card No.	FORTRAN Symbol	Variable Name	Units	Definition
	DTHETA	$\Delta\theta$	degrees	Increment for θ. (A typical value is .05)
	RLGTH	R_o	length	Characteristic container dimension
6	N	n		No. of equations
	IBOPT			Option for calculation of empty fraction β . If = 0, suppress calculation†; if = 0, calculate β
7	NN(1)			Two choices of no. of interface points of equal intervals of which
	NN(2)			the locations are desired.

†In case y_o is known from either another theory or experiments, one may proceed the calculation with a dummy mass fraction. In theoretical predictions, usually desired empty fraction is given, then we must use IBOPT = 1 to calculate β for each assumed y_o in the program, while the input y_o is an approximate guess.

TABLE III. OUTPUT OF PROGRAM SSHAPE

Printed Output

1. Input data ALPHA, KO, YO, BN, THETA, Y(1), Y(2)
2. THETA, XOR, ZOR, Y(1), Y(2)
3. GAMMA, PHI, ALPHA, YO, KO
4. Extrapolated YO (non-dimensional)
5. VU1, VU2, VU, VL, VT
6. BETA
7. G, RS, S, XR, XY
8. K2F, FRR2 (interface curvature n_F and $\frac{d^2 F}{d_R 2}$ on F at the contact point)
9. K2B, ZRR2 (wall curvature n_B and $\frac{d^2 Z}{d_R 2}$ on wall at the contact point)
10. CGAM (Γ)
11. Dimensional YO
12. XRB, XYB
13. ZCGT, BZCGU, ZCG

FORTRAN Symbol	Variable Name	Units	Description
S_i	S	length	Arc length, i = 1 at interface center point
G_i	G	length square	Sum of curvatures squared, i = 1 at interface center point
RS_i	R_s	length	dR/dS, i = 1 at interface center point

TABLE III. OUTPUT OF PROGRAM SSHAPE (Cont'd)

FORTRAN Symbol	Variable Name	Units	Description
CGAM	Γ	non-dim.	Contact point constant
XR	XR	length	Horizontal distance from container vertical axis to liquid-vapor interface with origin at interface center point
XY	XY	length	Vertical distance from x-axis to liquid vapor interface with origin at interface center point
XRB	XRB	length	x-coordinates of container boundary with origin at interface center point
XYB	XYB	length	y-coordinate of container boundary with origin at interface center point
ZCGT	Z_{CGT}	length	Container c.g.
BZCGU	βZ_{CGU}	length	Mass fraction times distance of ullage c.g. upward from interface center point
ZCG	Z_{CG}	length	Distance of liquid c.g. upward positive from interface center point

TABLE IV. DEFINITION OF TERMS IN SSHAPE

Some of the program symbols which were not defined previously.

FORTRAN Symbol	Variable Name	Units	Definition
XOR	X/R_o	non-dim.	Horizontal distance from container vertical axis to interface point
ZOR	Z/R_o	non-dim.	Vertical distance (downward positive) from x-axis to interface point, origin at top center.
Y(1)	y	non-dim.	Radial distance from top center to interface point
Y(2)	y'	non-dim.	First derivative of y with respect to θ
GAMMA	γ	degrees	Angle measured from vertical and tangent to $y(\theta)$ at contact point
PHI	ϕ	degrees	Angle measured from vertical and tangent to $y_B(\theta)$ at contact point
VU1		length cube	Vapor volume to point of contact
VU2		length cube	Vapor volume from point of contact to boundary
VU		length cube	Vapor volume
VL		length cube	Liquid volume
VT		length cube	Total container volume

TABLE V. EXAMPLE KEYPUNCH FORM FOR SSHAPE

COMPUTATIONS LABORATORY

SOUTHWEST RESEARCH INSTITUTE

PROBLEM Example Keypunch Form

PAGE OF

PROGRAMMER

DATE 8/20/69

FORTRAN STATEMENT

```
   NP - FØRMAT ( 2I5 )
    3

   YBX(I), I=1,NP - FØRMAT ( 8F10.0 )
0.        0.68      0.68

   YBZ(I), I=1,NP - FØRMAT ( 8F10.0 )
2.04      2.04      0.

ALPHA,KO,YO,BN,BETAD,DKO,DBC,DYO - FØRMAT ( 8F10.0 )
2.        0.        0.45      100.      0.4       0.2       0.005     0.05

THETA,    DTHETA,   RLGTH - FØRMAT ( 8F10.0 )
0.        0.05      0.68

    N,   IBØPT - FØRMAT ( 2I5 )
    2    0

NN(1),   NN(2) - FØRMAT ( 2I5 )
   23    0
```

APPENDIX II

A THEORY† FOR LOW-GRAVITY FUEL SLOSHING IN AN ARBITRARY AXISYMMETRIC RIGID TANK

Wen-Hwa Chu

†This theory and main results have already been published as ASME paper 70-APM-EEE to appear in the Journal of Applied Mechanics and contains major extensions and modifications of Technical Report No. 8. It is included for convenience of the sponsor and the readers of this report.

Introduction

THE behavior and consequences of fuel sloshing in rockets under a high effective gravity were recognized problems which have been quite well understood [1-3] [1] The problem of low-gravity fuel sloshing, characterized by the significant role of interfacial tension, is now a subject of importance for application to coasting rockets or orbital stations

The equilibrium behavior of fluids at zero and/or low gravity has been studied in references [4-7] The theoretical determination of an equilibrium interface shape is nonlinear and requires a trial and error procedure for a given contact angle [5, 6]

Satterlee and Reynolds [8] have successfully solved the free sloshing problem in cylindrical containers under low gravity and formulated a variational principle for this purpose Yeh [9], using a similar approach, solved the free and forced sloshing problem under low-gravity conditions, without force and moment or an equivalent mechanical model Dodge and Garza [10, 11] performed force measurements under simulated low-gravity conditions and predicted forces and moment for circular cylindrical tanks under lateral (translational) motion The equivalent spring-mass model was given in [10] Additional work by Dodge and Garza for other special tanks was given in [12, 13] A finite-difference approach with application to a hemispherically bottomed cylindrical tank and spheroidal tanks was given by Concus, Crane, and Satterlee in [14, 15]

These investigations indicate a need of a program for a general axisymmetric tank A preliminary study on liquid sloshing in an arbitrary axisymmetric tank was reported in [16], but it is limited to translational oscillations It is the object of the present paper to present a seminumerical approach for an arbitrary axisymmetric tank with simplified force and moment calculations and the resultant mechanical model for both pitching and translational oscillations A general computer program utilizing Winslow method [17] will be completed to obtain sloshing frequencies, slosh mass, and mass-height, for which a brief description is given in Appendix A

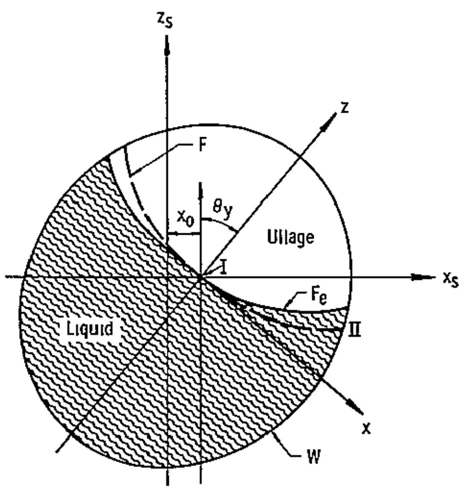


Fig 1 Some Nomenclatures

[1] Numbers in brackets designate References at end of paper

Contributed by the Applied Mechanics Division for publication (without presentation) in the JOURNAL OF APPLIED MECHANICS

Governing Equations

Assuming irrotational incompressible flow, there is a space-fixed velocity potential[2] ϕ satisfying the Laplace equation in both space-fixed and tank-fixed coordinates

$$\nabla_s^2\phi = 0, \ \nabla^2\phi = 0,$$

$$\nabla_s^2 = \frac{\partial^2}{\partial x_s^2} + \frac{\partial^2}{\partial y_s^2} + \frac{\partial^2}{\partial z_s^2}, \ \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (1)$$

As in thin airfoil theory, the velocity potential can be obtained by imposing boundary conditions on the initial or mean position, but the hydrostatic pressure due to gravity possesses components along both the tank axis z and the lateral axis x, Fig 1, for pitching oscillations The linearized Bernoulli's equation states

$$p - p_I + \rho\frac{\partial\phi}{\partial t} + \rho g(z - x\theta_y) = 0 \quad (2)^3$$

and

$$p - p_{uI} + \rho_u\frac{\partial\phi_u}{\partial t} + \rho_u g(z - x\theta_y) = 0 \quad (3)$$

for the liquid and the ullage, respectively, and p_I, p_{uI} are constants

Boundary Conditions

The linearized interface kinematic condition states that

$$\frac{\partial h}{\partial t} \cong \frac{\partial\phi}{\partial n} \quad \text{on } F \quad (4)^4$$

The interface dynamic condition states that

$$-p_- + p_+ = \sigma\kappa = \sigma\kappa_0 + \sigma\kappa' \quad \text{on } F_e \quad (5)$$

For the "mean" interface location f (in general, $p_I = p_I{}^\circ + p_I'$, $p_I{}^\circ$, p_I' being constants),

$$-\sigma\kappa_0 + (\rho - \rho_u)gf - (p_I{}^\circ - p_{uI}{}^\circ) = 0 \quad \text{on } F \quad (6)$$

where the curvature of the mean interface, κ_0, is axisymmetric and

$$\kappa_0 = \frac{1}{r}\frac{\partial}{\partial r}\left\{\frac{r\dfrac{\partial f}{\partial r}}{\sqrt{1 + \left(\dfrac{\partial f}{\partial r}\right)^2}}\right\} = \frac{1}{r}\frac{\partial f}{\partial s} + \left(\frac{\partial r}{\partial s}\frac{\partial^2 f}{\partial s^2} - \frac{\partial f}{\partial s}\frac{\partial^2 r}{\partial s^2}\right) \quad (6a)$$

Equation (6) holds for $r = 0$, thus

$$p_I{}^\circ - p_{uI}{}^\circ = -2\left(\frac{\partial^2 f}{\partial r^2}\right)_{II}$$

The linearized interface dynamic condition is then

$$-(p_u' - p_{uI}') - \sigma\kappa' + \rho\frac{\partial\phi}{\partial t} - \rho_u\frac{\partial\phi_u}{\partial t} + (\rho - \rho_u)gh\frac{\partial r}{\partial s} - (\rho - \rho_u)gx\theta_y = 0 \quad (7)^5$$

[2] For example, $\partial\phi/\partial x$ gives velocity component in x-direction with respect to space-fixed coordinates

[3] Chu, W H, "Free Surface Condition for Sloshing Resulting From Pitching and Some Corrections," *ARS Journal*, Vol 30, Nov 1960, pp 1093-1094

[4] A vertical interface description was used but is only successful when $(dF/dR)^{-1}$ does not vanish on the interface

[5] For sinusoidal oscillations and $m = 1$, $h = 0$, $\phi = \phi_u = 0$, $x = 0$, and $\kappa' = 0$ at point I, thus $p_I' - p_{uI}' = 0$ For other m values, $-p_I' + p_{uI}{}^\circ = \sigma\kappa_I'$, which will be omitted until needed

where the perturbation curvature for $\cos(m\theta)$ variation is to the first order

$$\kappa' = \frac{\partial^2 h}{\partial s^2} + \frac{1}{r}\frac{\partial r}{\partial s}\frac{\partial h}{\partial s} + \frac{1}{r^2}\frac{\partial^2 h}{\partial \theta^2} + \left[\left(\frac{1}{r}\frac{\partial f}{\partial s}\right)^2 + \left(\frac{\partial r}{\partial s}\frac{\partial^2 f}{\partial s^2} - \frac{\partial f}{\partial s}\frac{\partial^2 r}{\partial s^2}\right)^2\right] h \quad (7a)^6$$

m being unity for lateral excitation of a rigid tank. At point I, the origin, $h = 0$, $\phi = 0$, $\kappa' = 0$, and thus $p_I = p_{uI}$. For most analyses, $p_u = 0$ was assumed. We shall assume the impulsive pressure in the ullage is negligible, i.e., $\phi_u = 0$. Then, for sinusoidal oscillations with h, ϕ proportional to $\cos(\omega t)$ and $\sin(\omega t)$, respectively, equations (7), (7a), and (4) yield

$$-\left\{\frac{1}{R}\frac{\partial}{\partial s}\left(R\frac{\partial H}{\partial s}\right) - \frac{m^2}{R^2}H + GH\right\} + N_{B_e}H\frac{\partial R}{\partial s} + \theta_y N_{B_e} R\cos\theta + \Omega^2\Phi = 0 \qquad \text{on } F \quad (7b)^7$$

where

$$G = \left(\frac{1}{R}\frac{\partial F}{\partial s}\right)^2 + \left(\frac{\partial R}{\partial s}\frac{\partial^2 F}{\partial s^2} - \frac{\partial F}{\partial s}\frac{\partial^2 R}{\partial s^2}\right)^2 = \frac{1}{R^2}\frac{F_R{}^2}{(1 + F_R{}^2)} + \left[\frac{F_{RR}}{(1 + F_R{}^2)^{3/2}}\right]^2 \quad (7c)$$

It is noted that, for lateral oscillations, $m = 1$, solutions are proportional to $\cos\theta$.

The boundary condition on the wall is that the relative normal velocity be zero, i.e., with $\cos(n, x) = \frac{\partial x}{\partial n}$ and $\cos(n, z) = \frac{\partial z}{\partial n}$,

$$\frac{\partial \phi}{\partial n} = x_0 \frac{\partial x}{\partial n} \quad (8)$$

and

$$\frac{\partial \phi}{\partial n} = \theta_y\left(z\frac{\partial x}{\partial n} - x\frac{\partial z}{\partial n}\right) \quad (9)$$

for translational and pitching oscillations, respectively.

In addition, there is an interface contact point condition which takes the form [8, 9, and 15]

---

[6] The instantaneous interface can be described by the position vector $\mathbf{r}/a = [R - HF_s]\cos\theta\hat{i} + [R - HF_s]\sin\theta\hat{j} + [F + HR_s]\hat{k}$, $R_s = dR/ds$, $F_s = dR/ds$. The curvatures can then be derived from the fundamental magnitudes [18] neglecting higher-order terms. The linearized total curvature is given by this equation, which was first derived in [15] by a different procedure.

[7] For sinusoidal oscillations, without loss of generality, x_0, θ_y, ϕ are assumed to be proportional to $\sin(\omega t)$ while h is proportional to $\cos(\omega t)$.

Nomenclature

- a = reference length, say, maximum radius of tank
- dA = $rdrd\theta$
- dA = dA/a^2 (a scalar)
- dS = 3-D surface element, e.g., $rd\theta ds$
- dS = dS/a^2, nondimensional surface element (a scalar)
- F = equilibrium (mean) interface of f/a
- F_e = instantaneous interface
- F_H = horizontal force defined by equation (19)
- F_R = (dF/dR), slope of F in the generatrix plane
- F_x = x-component of force on tank
- f = equilibrium (mean) interface elevation measured along vertical axis
- g = gravitational acceleration
- H = amplitude of h/a, nondimensional slosh height
- h = interface perturbation normal to equilibrium interface
- h_0 = reference length, say, depth of liquid at center of tank
- M_0, I_0 = rigid mass and moment of inertia of mechanical model
- M_L = liquid mass
- M_y = pitching moment about y-axis
- m_k = kth slosh mass
- N_B = Bond number, $\rho a^2 g/\sigma$
- n = outer normal
- n_0 = n/a, nondimensional normal distance
- p = pressure
- p_I = equilibrium liquid pressure at origin—a constant
- p_u = ullage pressure
- p_{uI} = equilibrium ullage pressure at origin—a constant
- R = r/a, nondimensional radius
- r, θ, z = tank fixed cylindrical coordinates
- s = arc length nondimensionalized by a
- s = arc length (a scalar)
- t = time
- V = volume of liquid divided by a^3
- V_l = liquid volume
- W = wall wetted by liquid
- W_e = instantaneous wetted wall below instantaneous interface, F_e
- x_0 = translational amplitude in x_s-direction
- x_s, y_s, z_s = space-fixed rectangular coordinates
- Γ = γa, nondimensional contact point constant
- γ = hysteresis coefficient or contact point constant
- $\Delta\rho$ = density difference, $\rho - \rho_u$
- δ_{ij} = Kronecker delta
- ϵ_I = sign of $n \cdot \hat{z}$, $\cos(n, z)$, or $\partial z/\partial n$
- θ_y = amplitude of pitching about y-axis
- κ = mean curvature
- κ' = perturbation of mean curvature
- λ_j = jth eigenvalue ($m = 1$)
- λ_{mj} = jth eigenvalue corresponds to mth circumferential mode
- ρ = liquid density
- ρ_u = density of ullage fluid (vapor or gas)
- σ = surface tension
- Φ = amplitude of nondimensional velocity potential, $\phi/\omega a^2$
- Φ_k = amplitude of nondimensional potential $\phi_k/\omega a^2$
- Φ_N = see equation (15)
- Φ° = amplitude of nondimensional potential $\phi^\circ/\omega a^2$
- ϕ = velocity potential
- ϕ_k = velocity potential of the kth natural mode
- ϕ' = additional velocity potential due to interface movement
- ϕ° = velocity potential of liquid with a frozen interface
- ψ = velocity potential of auxiliary eigenfunctions
- ψ_j = jth auxiliary characteristic function
- Ω^2 = $\rho a^3\omega^2/\sigma$, product of Bond number, $\rho a^2 g/\sigma$, and frequency parameter, $\omega^2 a/g$
- ω = frequency of oscillation
- ω_k = kth natural frequency

Subscripts

- $(\ )_I$ = $(\ )$ at vertex of equilibrium interface (origin)
- $(\ )_{II}$ = $(\ )$ at contact point in generatrix plane
- $(\ )_{cG}$ = $(\ )$ related to center of gravity
- $(\ )_e$ = effective value of $(\ )$
- $(\ )_F$ = $(\ )$ on F
- $(\ )_m$ = $(\ )$ associated with $\cos(m\theta)$ mode
- $(\ )_p$ = $(\ )$ related to pitching
- $(\ )_T$ = $(\ )$ related to translation
- $(\ )_W$ = $(\ )$ on W
- $(\ )_u$ = $(\ )$ related to ullage
- $(\ )_-$ = $(\ )$ just below interface
- $(\ )_+$ = $(\ )$ just above interface

$$\frac{\partial h}{\partial s} = \gamma h \quad \text{or} \quad \frac{\partial H}{\partial s} = \Gamma H \quad \text{at point II} \tag{10}$$

where γ may be a frequency-dependent constant However, if the contact angle remains constant, we can show that (see Appendix C[8])

$$\gamma = + \frac{1}{\sin\theta_c}\left\{\left|\frac{z_{rr}}{(1+z_r^2)^{3/2}}\right|_{W\,II} - \cos\theta_c\left|\frac{f_{rr}}{(1+f_r^2)^{3/2}}\right|_{II}\right\} \tag{10a}$$[9]

$$\Gamma = \gamma a \tag{10b}$$

where

$$z_r = \frac{dz}{dr}, f_r = \frac{df}{dr}, z_{rr} = \frac{d^2z}{dr^2}, f_{rr} = \frac{d^2f}{dr^2}, \quad z \text{ on } W, f \text{ on } F$$

Method of Solution

We shall decompose ϕ into two parts, ϕ' and ϕ° ϕ° is the velocity potential corresponding to a liquid contained by a rigid mean interface and the tank walls Therefore, it satisfies the Laplace equation and the nonhomogeneous boundary condition on the contour, equation (8) for translation and equation (9) for pitching on F_e and W_e It is noted that

$$\phi_T^0 = x_0 x \tag{11}$$

while ϕ_p° can be constructed numerically

ϕ' is the perturbed velocity potential due to sloshing which is governed by the Laplace equation, the zero normal velocity condition at the wall and the resultant interface condition The first two are satisfied by expansion in normal modes and the last by equations (15) and (16) In this paper, the normal modes are determined by expansion into a set of auxiliary characteristic functions[10] ψ_j, which is orthogonal on the curved interface, and satisfies the Laplace equation and the zero normal velocity wall condition [11] The interface condition governing normal modes is second order [see equation (7b)] subject to zero H at center and the contact point condition, equation (10a) at wall The former is satisfied as ψ_j vanishes along tank center line The interface equation is approximately satisfied by a modified Galerkin method (the strict Galerkin method is given in [19]), which is illustrated in Appendix B In this method, the contact point condition was imposed on the whole series, not term by term The numerical results shown later substantiate the method used

The force and moment are obtained by integration of pressure, not only on the wall, but also on the interface since the direct surface tension force and moment on the tank is equivalent to those on the interface due to pressure, assuming the interface inertia is negligible, as well as the interface mass To put results in the mechanical model form, the divergence theorem has been most useful (with some manipulations)

Analytical Results

Free Oscillations. For free oscillations, the natural mode ϕ_k is expanded into a truncated series of the auxiliary eigenfunctions, i e,

[8] Appendix is not in order of mention

[9] An equivalent form was first derived in reference [15]

[10] For direct application of the Winslow method [17], we impose the simpler normal derivation condition, $\partial\psi_j/\partial n_0 = \lambda\psi_j$, on F and used the well-known influence coefficient technique to determine the eigenvector ψ_j on the intersurface, the eigenvalue λ_j

[11] Strictly speaking, the boundary conditions are imposed in Winslow method, while the solution not proven to satisfy these conditions is correct on physical grounds (see Appendix A)

$$\Phi_k = \frac{\phi_k}{\omega a^2} = \sum_{j=1}^{J_{mz}} c_{kj}\psi_{m_j}\cos(m\theta),\ \psi_j = \psi_{m_j}\cos(m\theta) \tag{12a, b}$$

$$H_k = \frac{h_k}{a} = -\sum_{j=1}^{J_{mz}} c_{kj}\psi_{m_j}\cos(m\theta) \tag{12c}$$

c_k, is the kth eigenvector of the following matrix equation obtained by a modified Galerkin method (Appendix B) from integrating the nondomensional equation (7b) with $\theta_y = 0$ and weighting function $\psi_{m_i}dS/\alpha_{m_i}^2$

$$\left\{-\Gamma[\nu_{m_{ij}}] + [\gamma_{m_{ij}}] + m^2[\epsilon_{m_{ij}}] + \frac{\Delta\rho}{\rho}N_B[\beta_{m_{ij}}] - \Omega^2[\Delta_{m_{ij}}]\right\}\{c_j\} = 0,\ i, j = 1 \text{ to } J_{mz} \tag{13}$$

where

$$\beta_{m_{ij}} = \frac{\lambda_{m_{ij}}}{\alpha_{m_i}^2}\int_F \frac{\psi_{m_i}\psi_{m_j}\epsilon_1}{\sqrt{1+F_R^2}}dS,\ \epsilon_1 = \operatorname{sgn}\left(\frac{\partial R}{\partial s}\right) \tag{13a}$$[12]

$$\epsilon_{m_{ij}} = \frac{\lambda_{m_j}}{\alpha^2_{m_i}}\int_F \frac{1}{R^2}\psi_{m_i}\psi_{m_j}dS \tag{13b}$$

$$\nu_{m_{ij}} = \frac{2\pi\lambda_{m_{ij}}}{\alpha_{m_i}^2}[R\psi_{m_i}\psi_{m_j}] \tag{13c}$$

$$\gamma_{m_{ij}} = \frac{\lambda_{m_j}}{\alpha_{m_i}^2}\left\{\int_F [1+F_R^2]^{-1}\frac{d\psi_{m_i}}{dR}\frac{d\psi_{m_j}}{dR}dS - \int_F G\psi_{m_i}\psi_{m_j}dS\right\} \tag{13d}$$

$$\Delta_{m_{ij}} = \frac{1}{\alpha_{m_i}^2}\int_F \psi_{m_i}\psi_{m_j}dS = \delta_{ij} \tag{13e}$$

$$\alpha_{m_i}^2 = \int_F \psi_{m_i}^2 dS \tag{13f}$$

and $m = 1$ for lateral excitation of a rigid tank

Forced Oscillations Let

$$\phi' = -\omega a^2\sum_{k=1}^{K_{mz}} d_k\Phi_k,\ d_k = \frac{\bar{d}_k\Omega^2}{\Omega_k^2 - \Omega^2},\ \Phi_k = \sum_{j=1}^{J_{mz}} c_{kj}\psi_j \tag{14a, b, c}$$

in order to satisfy the interface condition that

$$\sum_{k=1}^{K_{mz}} \bar{d}_k(\Omega_k^2 - \Omega^2)\Phi_k = -\Omega^2\Phi^\circ + \epsilon_2 N_B\frac{\Delta\rho}{\rho}\frac{x}{a}\theta_y \equiv \Phi_N\Omega^2 \tag{15}$$

$\epsilon_2 = 0$ for translational oscillation, $\epsilon_2 = 1$ for pitching oscillation We have by the Galerkin procedure [19, equation (115 5), p 435]

$$\sum_{k=1}^{K_{mz}} \bar{d}_k\int_F \Phi_k(-H_l)dS = \int_F \Phi_N(-H_l)dS,\ l = 1 \text{ to } K_{mz} \tag{16}$$

A weighting function of H_l was used taking advantage of the biorthogonal relation, equation (17) This assures the first K_{mz} components of the error as series in ϕ_k be zero $\bar{d}_k$ can be solved from equation (16) by matrix inversion There is no need of storing information of ψ_j inside the fluid domain as only the force and moment are of interest It is noted in the limit [8 and 9]

$$\int_F \Phi_k(-H_l)dS = \delta_{kl}\int_\Gamma \Phi_l(-H_k)dS \tag{17}$$[13]

[12] The orthogonality property of ψ_j, thus, ψ_{m_j}, can be easily proved [16] as in the high-G case $\cos^2(m\theta)$ in the integrand are uniformly omitted as they contribute the same factor, 1/2

[13] This is referred to as biorthogonal relation

then,

$$\bar{d}_l = \frac{\int_F \Phi_N(-H_l)dS}{\int_F \Phi_l(-H_l)dS} \tag{18}$$

which was utilized in proving that a unique spring-mass system exists for both pitching and translation

Force and Moment The force and moment exerted by a spring-mass system, Fig 2, without damping were written in the following form [20] and remain valid if contributions due to direct surface tension are included

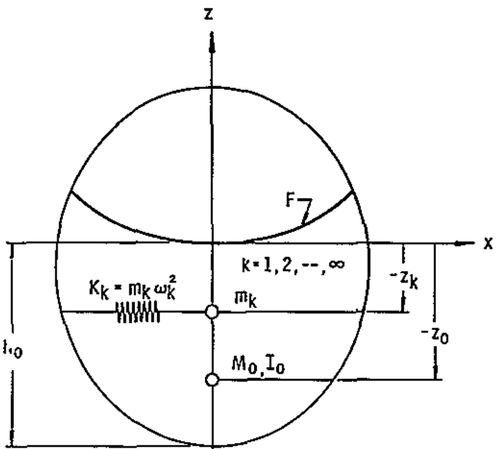


Fig 2 Equivalent mechanical model

$$F_H = F_x - M_F g\theta_y \tag{19}$$

$$F_{H_T} = x_0\omega^2 M_F \left\{1 + \sum_{k=1}^{\infty} \frac{m_k}{M_F} \frac{1}{\left(\frac{\omega_k^2}{\omega^2} - 1\right)}\right\} \tag{20}$$

$$M_{y_T} = x_0\omega^2 M_F h_0 \left\{\frac{z_{CG}}{h_0} + \sum_{k=1}^{\infty} \frac{m_k}{M_F}\left(\frac{z_k}{h_0} + \frac{g}{h_0\omega^2}\right) \frac{1}{\left(\frac{\omega_k^2}{\omega^2} - 1\right)}\right\} \tag{21}$$

$$F_{H_P} = \theta_y\omega^2 M_F h_0 \left\{\frac{z_{CG}}{h_0} + \sum_{k=1}^{\infty} \frac{m_k}{M_F}\left(\frac{z_k}{h_0} + \frac{g}{h_0\omega^2}\right) \frac{1}{\left(\frac{\omega_k^2}{\omega^2} - 1\right)}\right\} \tag{22}$$

$$M_{y_P} = \theta_y\omega^2 M_F h_0^2 \left\{\frac{I_F}{M_F h_0^2} + \sum_{k=1}^{\infty} \frac{m_k}{M_F}\left(\frac{z_k}{h_0} + \frac{g}{\omega^2 h_0}\right)^2 \frac{1}{\left(\frac{\omega_k^2}{\omega^2} - 1\right)}\right\} + \theta_y g M_F h_0 \left(\frac{z_{CG}}{h_0}\right) \tag{23}$$

with rigid mass m_0, its location z_0, and moment of inertia I_0 given by

$$\frac{m_0}{M_F} = 1 - \sum_{k=1}^{\infty} \frac{m_k}{M_F} \tag{24}$$

$$\frac{z_0}{h_0} = \frac{1}{\frac{m_0}{M_F}}\left[\frac{z_{CG}}{h_0} - \sum_{k=1}^{\infty} \frac{z_k}{h_0}\frac{m_k}{M_F}\right] \tag{25}$$

$$\frac{I_0}{M_F h_0^2} = \frac{I_F}{M_F h_0^2} - \left(\frac{m_0}{M_F}\frac{z_0^2}{h_0^2} + \sum_{k=1}^{\infty} \frac{m_k}{M_F}\frac{z_k^2}{h_0^2}\right) \tag{26}$$

The force due to liquid pressure and direct surface tension is

$$F_x = \int_{W_e+F_e} p\,\frac{\partial x}{\partial n}\,dS \tag{27}$$ [14]

and the moment due to liquid pressure and direct surface tension is

$$M_y = \int_{W_e+F_e} p\left(z\frac{\partial x}{\partial n} - x\frac{\partial z}{\partial n}\right) dS \tag{28}$$

it can be shown that

$$\frac{m_k}{M_F} = \bar{d}_{kT} f_k' \frac{1}{V} \tag{29}$$

where

$$f_k' = \sum_{j=1}^{\infty} c_{kj} \int_F \lambda_j \psi_j \frac{x}{a}\,dS \tag{29a}$$

$$\bar{d}_{kT} = \frac{1}{\beta_k^2}\int_{W+F} \Phi_k \frac{\partial x}{\partial n}\,dS \cong \frac{1}{\beta_k^2}\sum_{j=1}^{J_{mx}} c_{kj}\lambda_j \int_F \frac{x}{a}\psi_j\,dS \tag{29b}$$ [15]

$$V = V_L/\rho a^3,\ \beta_k^2 = \int_F \Phi_k(-H_k)dA \tag{29c, d}$$

and

$$\frac{z_k}{a} = \frac{1}{\frac{m_k}{M_F}}\left(\frac{l_k'}{V}\right) \tag{30}$$

where

$$l_k' = \bar{d}_{kT}\sum_{j=1}^{\infty} c_{kj}\mu_j \tag{30a}$$

$$\mu_j = \int_{F+W} \psi_j \left(\frac{z}{a}\frac{\partial x}{\partial n} - \frac{x}{a}\frac{\partial z}{\partial n}\right) dS \tag{30b}$$

and that

$$I_F = M_F a^2 I_F^*,\ I_F^* \cong \frac{1}{V}\int_{W+F} \Phi_p^\circ \left(\frac{z}{a}\frac{\partial r}{\partial n} - \frac{x}{a}\frac{\partial z}{\partial n}\right) dS \tag{31a, b}$$

In deriving the mechanical model, ρ_u has been set to zero A simple modification can be made for small ullage density by using

$$N_{B_e} = \frac{\Delta\rho}{\rho} N_B$$

the effective Bond number instead of the Bond number based on the density of the liquid, provided that the dynamic pressure due

[14] It is important to note that great simplification in results, when reduced to the mechanical model for both translation and pitching, is achieved by consideration of balance of forces acting on the thin interface as a free body The force and moment on the tank due to direct surface tension must be equal to those acting on the interface by liquid pressure These were not included in reference [15]

[15] For finite J_{mx}, it was found that $\bar{d}_{kT}$, determined by matrix inversion of equation (16) without using biorthogonal relation, yields results in better agreement with Dodge's theory [12] than equation (29b) which is correct in the limit

to ullage motion is negligible

It is noted that to prove the mechanical model for pitching motion frequent application of divergence theorem combined with differential properties of the auxiliary characteristic functions and the coordinate functions x and z were used. For brevity, the details are not included in this paper

Numerical Examples The present computer program has been checked out for several examples in Appendix III. These include cylindrical tanks, spherical tanks, and spheroidal tanks. Good agreement of first spring-mass with known theory or experiments was obtained with as few as 11 or 12 interface net points and the first seven auxiliary characteristic functions. Finer nets may yield higher accuracies, especially for the higher modes

Machine Time For a 12×18 mesh and a 12×12 mesh, the CDC-6600 central process time is a little over 2 min, while for the 23×34 mesh it is about 21 min. Most of the computing time was expended for the generation of influence coefficients, each of which is a Neumann problem. However, the influence coefficient method may be more convenient than the inversion of a large matrix, if not faster. No computer running time was reported in references [14 and 15], which finds natural modes by finite-differences and (partial) matrix inversion

Conclusion

It seems that the present method yielded a practical way of computing the fundamental natural frequency, the first slosh mass, and its location. Higher masses and locations are usually not needed for design purposes and can be obtained by using finer meshes and longer machine time. A computer program utilizing triangular meshes and Winslow method [17] has been successfully employed and is expected to be completed in the near future for the titled problem. However, the present logical diagram may be limited to a convex axisymmetric tank for good accuracy

Acknowledgments

The author would like to take this opportunity to thank D. R. Saathoff, A. F. Mueller, and R. Gonzales for writing the computer programs, and Dr. Franklin T. Dodge for his helpful discussions. The present research is supported by NASA Contract NAS8-20290, George C. Marshall Space Flight Center

References

1 Abramson, H. N., ed., "The Dynamic Behavior of Liquids in Moving Containers," NASA Office of Scientific and Technical Information SP-106, Washington, 1966

2 Abramson, H. N., "Dynamic Behavior of Liquids in Moving Containers," *Applied Mechanics Reviews*, Vol 16, No 7, July 1963, pp 501–506

3 Cooper, R. M., "Dynamics of Liquids in Moving Containers," *ARS Journal*, Vol 30, Aug 1960, p 725

4 Neu, J. T., and Good, R. J., "Equilibrium Behavior of Fluids in Containers at Zero-Gravity," *AIAA Journal*, Vol 1, No 4, Apr 1963, pp 814–819

5 Satterlee, H. M., and Chin, J. H., "Meniscus Shape Under Reduced-Gravity Conditions," Symposium on Fluid Mechanics and Heat Transfer Under Low Gravitational Conditions, June 1965

6 Hastings, L. J., and Rutherford, R., III, "Low Gravity Liquid-Vapor Interface Shapes in Axisymmetric Containers and a Computer Solution," NASA TM X-53790, Oct 1968

7 Petrash, D. A., and Otto, E. W., "Studies of the Liquid Vapor Interface Configuration in Weightlessness," ARS Space Power System Conference, Santa Monica, Calif., Paper No 2514-62, 1962

8 Satterlee, H. M., and Reynolds, W. C., "The Dynamics of the Free Liquid Surface in Cylindrical Containers Under Strong Capillary and Weak Gravity Conditions," TR LG-2, Department of Mechanical Engineering, Stanford University, May 1, 1964

9 Yeh, G. C. K., "Free and Forced Oscillations of a Liquid in an Axisymmetric Tank at Low-Gravity Environments," Journal of Applied Mechanics, Vol 34, No 1, Trans ASME, Vol 89, Series E, Mar 1967, pp 23–28

10 Dodge, F. T., and Garza, L. R., "Experimental and Theoretical Studies of Liquid Sloshing at Simulated Low Gravities," Technical Report No 2, Contract No NAS8-20290, Control No DCN 1-6-75-00010, SwRI Project 02-1846, Southwest Research Institute, Oct 1966, see also Journal of Applied Mechanics, Vol 34, No 3, Trans ASME, Vol 89, Series E, Sept 1967, pp 555–562

11 Dodge, F. T., and Garza, L. R., "Simulated Low-Gravity Sloshing in a Cylindrical Tank Including Effects of Damping and Small Liquid Depth," *Proceedings, 1968 Heat Transfer and Fluid Mechanics Institute*, Stanford University Press, 1968, pp 67–69

12 Dodge, F. T., and Garza, L. R., "Simulated Low-Gravity Sloshing in Spherical Tanks and Cylindrical Tanks With Inverted Ellipsoidal Bottoms," Technical Report No 6, Contract NAS8-20290, Control No DCN 1-6-75-00010, SwRI Project No 02-1846, Southwest Research Institute, Feb 1968

13 Dodge, F. T., and Garza, L. R., "Slosh Force, Natural Frequency, and Damping of Low-Gravity Sloshing in Oblated Ellipsoidal Tanks," Technical Report No 7, Contract No NAS8-20290, Control No DCN 8-75-00043(IF), SwRI Project No 02-1846, Southwest Research Institute, Feb 1969

14 Concus, P., Crane, G. E., and Satterlee, H. M., "Low Gravity Lateral Sloshing in Hemispherically Bottomed Cylindrical Tanks," *Proceedings, 1968 Heat Transfer and Fluid Mechanics Institute*, Stanford University Press, 1968, pp 80–97

15 Concus, P., Crane, G. E., and Satterlee, H. M., "Small Amplitude Lateral Sloshing in Spheroidal Containers Under Low Gravity Conditions," NASA CR-72500, LMSC-A944673, Lockheed Missiles and Space Co., Sunnyvale, Calif., February 4, 1969

16 Chu, W. H., "Low Gravity Liquid Sloshing in an Arbitrary Axisymmetric Tank Performing Translational Oscillations," Technical Report No 4, Contract No NAS8-20290, Control No DCN 1-6-75-00010, SwRI Project No 02-1846, Southwest Research Institute, Mar 1967

17 Winslow, A. M., "Numerical Solutions of the Quasilinear Poisson Equations in a Nonuniform Triangular Mesh," *Journal of Computational Physics*, Vol 1, No 2, Nov 1966, pp 149–172

18 Wang, C. T., *Applied Elasticity*, McGraw-Hill, New York, 1953

19 Sokolnikoff, I. S., *Mathematical Theory of Elasticity*, 2nd ed McGraw-Hill, New York, 1956

20 Abramson, H. N., Chu, W. H., and Ransleben, G. E., Jr., "Representation of Fuel Sloshing in Cylindrical Tanks by an Equivalent Mechanical Model," *ARS Journal*, Dec 1961, pp 1697–1705

APPENDIX A

Brief Description of a Computer Program

The following steps of a computer program are briefly described

Construction of a Triangular Mesh The triangular mesh is generated as described in reference [17] except that a simple parallelogram is used as the logical diagram, Fig 3. For a cylindrical tank at Bond number 100, the physical diagram is shown in Fig

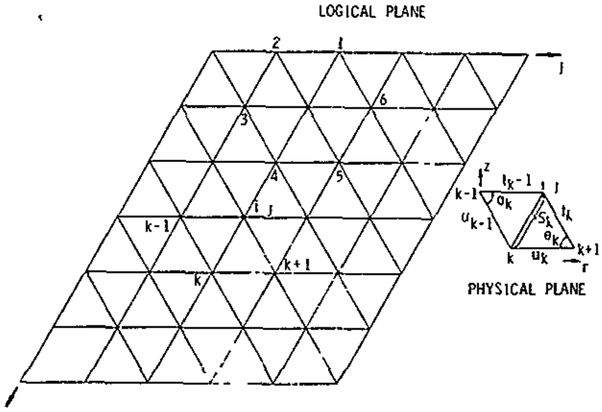


Fig 3 Simple logical diagram for triangular mesh

4(*a*) For a spheroidal tank ($e = 0.5$) at Bond number 5, a triangular mesh is shown in Fig 4(*b*) The lengths of the edge of the parallelogram can be adjusted for each individual case to yield desired triangular meshes A continuous wall needs to be broken into two parts for the logical diagram This only affects the local distribution of the triangular mesh and has shown to yield good results for a spheroidal tank, as well as a cylindrical tank

Construction of Auxiliary Characteristic Functions The characteristic functions ψ satisfy

$$\nabla^2\psi = 0 \tag{32}$$

$$\frac{\partial\psi}{\partial n_0} = 0 \quad \text{on } W \tag{33}$$

$$\frac{\partial\psi}{\partial n_0} = \lambda\psi \quad \text{on } F \tag{34}$$

ψ can be solved numerically with the constructed triangular mesh by Winslow method [17] Contact point is treated as one of the mesh points, as are the other boundary points [16] Hence, $\partial\psi/\partial n$ may be discontinuous at the contact point Zero contact angle cannot be constructed graphically but results of decrease mesh size give closer and closer approximations to the interface and would probably lead to the correct limiting value

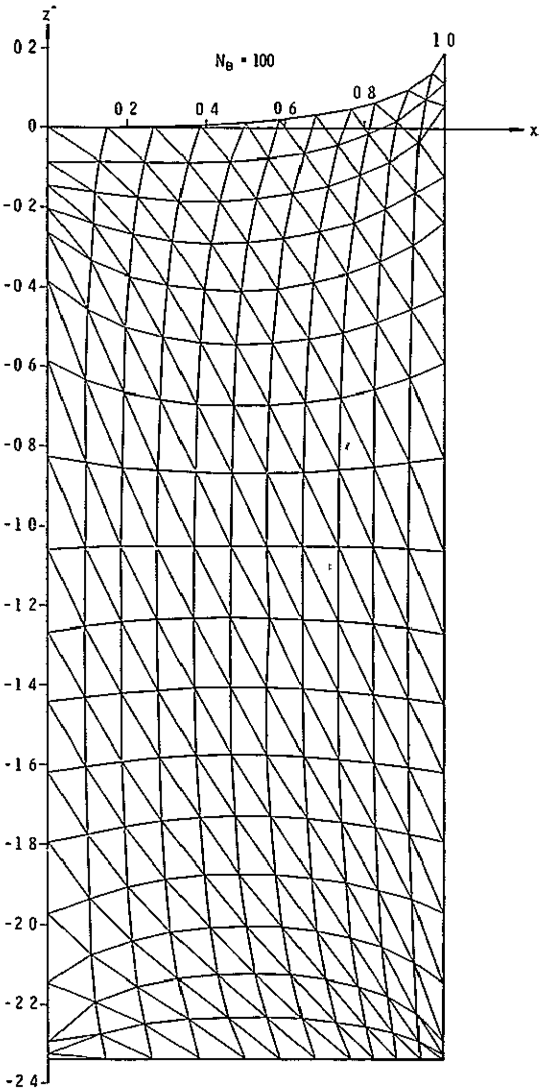


Fig 4(a) Physical diagram of triangular mesh—cylindrical tank

For an interior joint, ij, [$\psi = \psi_{i,j}$, $\psi_k = \psi_k(i, j)$, $r_k = r_k(i, j)$, $r = r_{i,j}$]

$$\sum_{k=1}^{6} \omega_k(\psi_k - \psi) - \frac{m^2}{r_{i,j}} A_{i,j}\psi = 0 \tag{35}$$

where

$A_{i,j}$ = area of the ijth dodecagon [17] inside the fluid domain
$r_{i,j}$ = radius of the ijth point

$$\omega_k = \tfrac{1}{2}(\lambda_k\bar{r}_k \cot\theta_k + \lambda_{k-1}\bar{r}_{k-1}\cot\sigma_k) \quad k = 1 \text{ to } 6 \quad (35a)^{17}$$

$$\bar{r}_k = \tfrac{1}{3}(r_{i,j} + r_k + r_{k+1}) \quad \lambda_k = 1 \qquad (35b, c)$$

θ_k, σ_k, Fig 3, can be expressed in terms of l_k, u_{k+1}, l_{k-1}, u_{k-1}, and s_k

For interface point,

$$\sum_{k=1}^{6} \omega_k(\psi_k - \psi) - \frac{m^2}{r_{i,j}} A_{i,j}\psi + \left(\frac{\partial\psi}{\partial n}\right)_{1,j}\left[\frac{1}{2}s_3 + \frac{1}{2}s_6\right]r_{i,j} = 0 \tag{36}$$

where

$$\lambda_6 = \lambda_1 = \lambda_2 = 0, \quad \lambda_3 = \lambda_4 = \lambda_5 = 1$$

To solve for the eigenfunctions on the interface, we use the influence coefficient method in which $(\partial\psi/\partial n)_{1,i} = 0$ except $(\partial\psi/\partial n)_{1,j} = 1$ for the jth column of the influence matrix A standard eigenvalue problem involving only the interface points, excluding ψ_{11} at $r = 0$, is needed to obtain the eigenvalues λ_j and eigenvectors ψ_j Knowing the jth eigenvector on the interface, the corresponding value of ψ_j on the wall can be easily solved numerically again by the method of overrelaxation

For ijth point on the tank wall

$$\sum_{k=1}^{6} \omega_k(\psi_k - \psi) - \frac{m^2}{r_{i,j}} A_{i,j}\psi = 0 \tag{37}$$

$\lambda_3 = \lambda_4 = \lambda_5 = 0$ and $\lambda_1 = \lambda_2 = \lambda_6 = 1$ on the bottom wall

$\lambda_4 = \lambda_5 = \lambda_6 = 0$ and $\lambda_1 = \lambda_2 = \lambda_3 = 1$ on the side wall

On center line, $r = 0$,

$$\psi = 0 \text{ for } m \geq 1$$
$$\frac{\partial\psi}{\partial r} = 0 \text{ for } m = 0 \tag{38}$$

$$\lambda_1 = \lambda_2 = \lambda_3 = 0 \text{ and } \lambda_4 = \lambda_5 = \lambda_6 = 1$$

At contact point $i = 1$, $j = j_{mx}$

$$\sum_{k=1}^{6} \omega_k(\psi_k - \psi) - \frac{m^2}{r_{i,j}} A_{i,j}\psi + \left(\frac{\partial\psi}{\partial n}\right)_{1\,j_{mx}}\left(\frac{1}{2}s_3\right)r_{i,j} = 0 \tag{39}$$

$$\lambda_3 = 1, \lambda_1 = \lambda_2 = \lambda_4 = \lambda_5 = \lambda_6 = 0$$

Calculation of Interface Shape A program based on the theory of reference [6] was written to generate the interface shape with desired net points for a given empty fraction or central depth and a

[16] This represents conservation of mass in the triangle containing contact point Experience shows that the contact point condition should not be imposed on ψ_j

[17] Note $(\lambda_{j-1/2})$ reference [17] $= \lambda_{j-1}$, $(\lambda_{j+1/2})$ reference [17] $= \lambda_j$

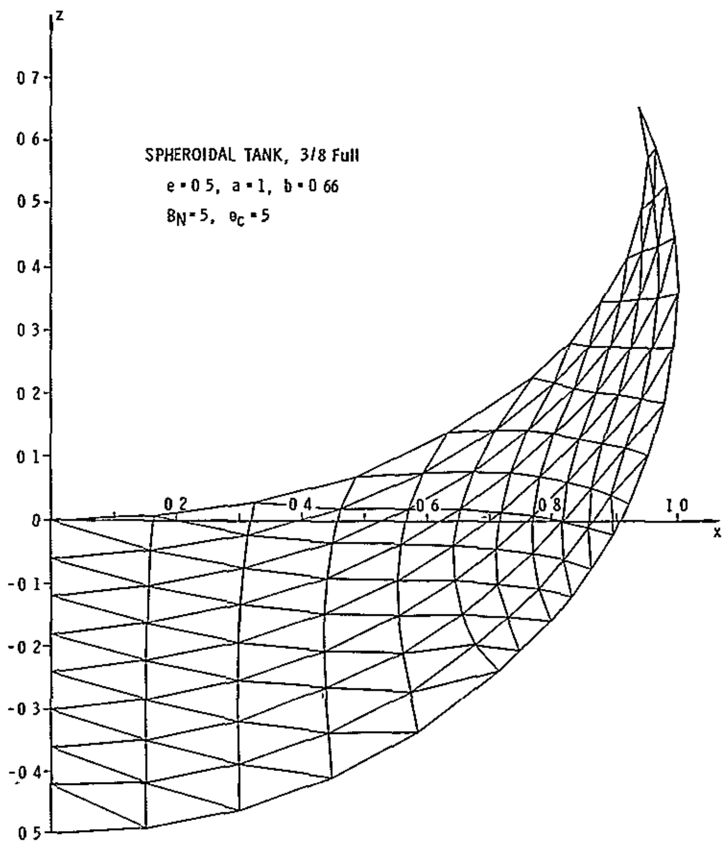


Fig 4(b) Triangular mesh for spheroidal tank at low gravity

given contact angle This program also calculates contact point constant Γ but limits the contact angle to possibly 2 deg or greater

Calculation of Natural Frequencies, Slosh Masses, and Their Location The remaining steps are relatively routine and, therefore, will not be described It is, however, remarked that trapezoidal rule and midpoint rules were employed conveniently in evaluating the integrals For some quantities, quadratic fittings were made, such as dF/dR, before entering the quadrature formulas

APPENDIX B

A Modified Galerkin Method

In the Galerkin method, it is generally assumed [19] that each of the coordinate functions of a complete set satisfies the same boundary conditions as the exact solution This condition seems to be a sufficient condition rather than a necessary condition since the series expansion as a whole may satisfy the prescribed conditions, not necessarily term by term Since the solution, if continuous, can be expanded into a convergent series containing the complete set, the failure of the error minimization process, if it occurs, must lie in the insufficient differentiability of the series Therefore, the present method as applied to a second-order ordinary differential equation performs integration by part, then imposes the remaining boundary condition[18] as illustrated by the following example

Let y be governed by the simple equation

$$\frac{d^2y}{dx^2} + \lambda^2 y = 0, \qquad \lambda \neq \frac{2m-1}{2}\pi \tag{40}$$

subject to

$$y = 0 \qquad \text{at } x = 0 \tag{41}$$

$$\frac{dy}{dx} = \Gamma_1 \qquad \text{at } x = 1 \tag{42}$$

We shall select a complete set in (0, 1) which satisfies $y = 0$ at $x = 0$ but $y \neq 0$ at $x = 1$ To be specific, we express

$$y = \sum_{m=1}^{\infty} C_m \sin(\lambda_m x) \tag{43}$$

where

$$\lambda_m = \frac{2m-1}{2}\pi$$

Then, the modified Galerkin procedure minimizes the error of the differential equation as follows

$$0 = \int_0^1 \sin(\lambda_n x)\left[\frac{d^2y}{dx^2} + \lambda^2 y\right] dx$$

$$= \left[\sin(\lambda_n x)\frac{dy}{dx}\right]_0^1 - \int_0^1 \lambda_n \frac{dy}{dx}\cos(\lambda_n x)dx$$

$$+ \int_0^1 \lambda^2 y \sin(\lambda_n x)dx = (-1)^{n-1}\Gamma_1$$

$$- \int_0^1 \lambda_n\lambda_m \sum_{m=1}^{\infty} C_m \cos(\lambda_n x)\cos(\lambda_m x)dx$$

[18] This may be a new technique to satisfy the boundary condition or conditions However, a general proof and extension remain to be done

$$+\int_0^1 \lambda^2 \sum_{m=1}^{\infty} C_m \sin(\lambda_n x)\sin(\lambda_m x)dx$$

$$= (-1)^{n-1}\Gamma_1 - \left(\frac{\lambda_n^2}{2} - \frac{\lambda^2}{2}\right) C_n, \qquad n = 1, 2, \quad , \infty \qquad (44)$$

Thus

$$C_n = \frac{2\Gamma_1(-1)^{n-1}}{(\lambda_n^2 - \lambda^2)} \qquad (45)$$

The exact solution of the problem is

$$y = \frac{\Gamma_1}{\lambda \cos \lambda} \sin(\lambda x) = \sum_{n=1}^{\infty} C_n' \sin(\lambda_n x) \qquad (46)$$

It is easy to show that the Fourier coefficient C_n' of the exact solution is equal to C_n, thus the modified Galerkin method yields the exact solution in the limit

The rate of convergence of the foregoing example ($\Gamma_1 \neq 0$) is only $1/n^2$ near the end, $x = 1$ This could be expected as dy/dx is zero term by term while with infinite terms converge to Γ_1 as a whole when $x \to 1$ Equation (13) in the text is analogously derived, however, the rate of convergence might be better as illustrated by results in Appendix III. It should be pointed out that at the high Bond numbers, each auxiliary characteristic function is a normal mode and, as Bond number decreases, more characteristic functions are required to represent each mode Only by actual computation is one to find out how many of them are sufficient for engineering purposes

APPENDIX C

Constant Contact Angle Condition

For small perturbed motions at the instantaneous contact point, Part 2, Fig 5.

$$\theta_2 = \theta_1 + \beta \qquad (47)$$

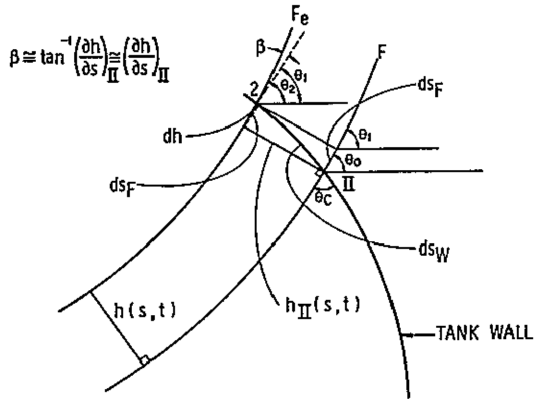
$$= \theta_0 + \left(\frac{\partial \theta_0}{\partial s}\right)_{II} dS_F + \left(\frac{\partial h}{\partial s}\right)_{II} \qquad (48)$$

With $\tan\theta_0 = \frac{df}{dr}, \frac{dr}{ds} = \frac{\epsilon_1}{\sqrt{1+f_r^2}}, \epsilon_1 = \operatorname{sgn}\left(\frac{dr}{df}\right)$, one has

$$\left(\frac{\partial \theta_0}{\partial s}\right)_{II} = \sec^2\theta_0 \left(\frac{d^2 f}{dr^2}\frac{\epsilon_1}{\sqrt{1+f_r^2}}\right)_{II} = \left[\frac{f_{rr}\epsilon_1}{(1+f_r^2)^{3/2}}\right]_{II} \qquad (49)$$

Similarly, at the instantaneous contact point 2,

$$\theta_{W_2} = \theta_{W\,II} + \left(\frac{\partial \theta_W}{\partial s_W}\right)_{II} ds_W \qquad (50)$$



$$ds_F = ds_W \cos\theta_c = h_{II} \cot\theta_c$$

$$ds_W = \frac{h_{II}}{\sin\theta_c}$$

Fig 5 Geometry for contact point condition

$$= \theta_{W\,II} + \left[\frac{z_{rr}\epsilon_2}{(1+z_r^2)^{3/2}}\right]_{W,II} ds_W \qquad (51)$$

where

$$\epsilon_2 = \operatorname{sgn}\left(\frac{dz}{dr}\right)$$

We shall assume constant contact angle independent of time t, i e ,

$$\theta_c = \theta_{W\,II} - \theta_0 = \theta_{W_2} - \theta_2 = \text{const} \qquad (52)$$

then, with $dS_W = \frac{h}{\sin\theta_c}$, $dS_F = dS_W \cos\theta_c$, Fig 6

$$\frac{f_{rr}\epsilon_1}{[1+f_r^2]^{3/2}} \cos\theta_c \frac{h}{\sin\theta_c} + \frac{\partial h}{\partial s} - \frac{z_{RR}\epsilon_2}{[1+z_R^2]^{3/2}}\frac{h}{\sin\theta_c} = 0 \quad \text{at II} \qquad (53)$$

i e ,

$$\frac{\partial h}{\partial s} = \gamma h \qquad (54)$$

where

$$\gamma = \frac{1}{\sin\theta_c}\left[\frac{z_{rr}\epsilon_2}{(1+z_r^2)^{3/2}} - \cos\theta_c \frac{f_{rr}\epsilon_1}{(1+f_r^2)^{3/2}}\right]_{II} \qquad (55)$$

For a convex tank with convex interface

$$\gamma = \frac{1}{\sin\theta_c}\left\{\left|\frac{z_{rr}}{(1+z_r^2)^{3/2}}\right|_{W\,II} - \cos\theta_c \left|\frac{f}{(1+f_r^2)^{3/2}}\right|_{II}\right\} \qquad (56)$$

APPENDIX III

NUMERICAL EXAMPLES

A. Flat Interface with High Bond Number in a Cylindrical Tank

$N_B = 1000$, $\frac{h_o}{a} = 2.34$, $a = 0.68$ with $\Gamma = 0$, a 12×18 mesh yielded $\frac{\omega_1^2 a}{g} = 1.85$ compared with 1.847 from exact theory (Ref. 1 of Appendix B).

$\frac{m_1}{m_F} = 0.193$ compared with 0.194 from high-G theory (Ref. 19 of Appendix B).

$z_1 = -0.729$ in. compared with -0.724 in. from high-G theory (Ref. 19 of Appendix B).

B. Flat Interface with Low Bond Number in a Cylindrical Tank $N_B = 10$, $\frac{h_o}{a} = 2.34$, $a = 0.68$ with $\Gamma = 0$, 12×18 mesh yielded $\frac{\omega_1^2 a}{g} = 2.15$ compared with 2.46 from exact theory. A finer mesh is required for better agreement.

C. Curved Interface with Low Bond Number $N_B = 100$, $\frac{h_o}{a} = 2.34$, $a = 0.68$, $\theta_c = 2°$ yielded $\frac{\omega_1^2 a}{g} = 1.860$, $\frac{m_1}{\rho a^3} = 0.442$, $z_1 \cong -0.724$ in. compared with theoretical value of $\frac{\omega_1^2 a}{g} = 1.777$, $\frac{m_1}{\rho a^3} = 0.438$, $z_1 \cong -0.73$ in. (Ref. 12 of Appendix B). It is noted that experimental value of $\frac{\omega_1^2 a}{g}$ for $\theta_c = 0°$ is around 1.78 to 1.80. Finer net may lead to better approximation.

D. Spherical Tank with Flat Interface $N_B = 1000$, $\frac{h_o}{a} = 1$, $a = 1$ with $\Gamma = 0$, 11×11 mesh yielded $\frac{\omega_1^2 a}{g} = 1.54$ compared with high-G sloshing value of $\frac{\omega_1^2 a}{g} = 1.54$.

E. Spherical Tank with "Folding" Interface $N_B = 10$, $\frac{h_o}{a} \cong 1.14$†, $a = 1$, $\theta_c = 4.96°$ ($\Gamma = -65.788$), 12×12 mesh yield $\frac{\omega_1^2 a}{g} = 1.469$, $\frac{m_1}{\rho a^3} = 0.604$, $z_1 = -0.742$ in. compared with Lockheed result (Ref. 16 of Appendix B) of $\frac{\omega_1^2 a}{g} = 1.507$ (for $\theta_c = 5°$). The first slosh mass is much larger than that of Lockheed, probably due to the inclusion of force directly attributed to surface tension in the present theory which yielded confirmed results for cylindrical tanks (Case 3). The value of z_1 would be the location of the center of the sphere if the contribution due to direct surface tension [i.e., the integral over F in Eq. (30b)] is neglected.

F. Spheroidal tank with Folding Interface Ellipticity $e = 0.5$, major axis $a = 1$, minor axis $b = 0.866$, $N_B = 5$, $\beta = 0.6263$ (3/8 full tank), $\frac{h_o}{a} \cong 0.4999$, $\theta_c = 5.06°$, ($\Gamma = -38.430$, $V_L = 1.35$).‡ 12×9 mesh (non-uniform on free surface) yielded $\frac{\omega_1^2 a}{g} = 1.114$††, $\frac{m_1}{\rho V_L} = 0.582$, $z_1 = -0.5841$ compared with Lockheed result of $\frac{\omega_1^2 a}{g} = 0.966$. It is noted that the value of z_1 puts the first mass under the tank bottom which should be examined by future experiments when available.

†Calculated for 3/4-full tank.

‡For $\theta_c = 5.06°$ the Γ used in ASME paper APM-EEE was in error and the correct value is given here as are the results.

††With 23×17 mesh, $\frac{\omega_1^2 a}{a} = 1.079$, $\frac{m_1}{\rho V_L} = 0.5736$, $z_1 = -0.4644$; and $\frac{\omega_2^2 a}{g} = 15.03$ while Lockheed's result for $\frac{\omega_2^2 a}{g}$ is 13.26. It is uncertain which results are more accurate. The value of Γ has not been given by Lockheed to facilitate the explanation of the differences.

APPENDIX IV.

LISTING AND INPUT SAMPLE OF TRIPOT

```
      PROGRAM TRIPOT(INPUT,OUTPUT,TAPE7,TAPE6)
C     NORMAL COORDINATES
C      02-1846-02  TRIANGULAR MESH GENERATOR AND SOLUTION OF
C     NORMAL MODES
      COMMON C(40,24,8),PHI(40,24),WB,EPS,ITA,N,M,WO,RHO,R(6),YT(7)
      DIMENSION X(40,24),Y(40,24),ITITLE(12)
      DIMENSION ELAMDA(6),SS(6),T(6),CT(6),CS(6),W(6),XT(7),A(6)
      DIMENSION CV(24),F(24,24),EVAL(24,24),EVEC(24,24),EVI(1,1)
      DIMENSION EIGVAL(24),NVAL(24),PHIS(24,24),PHIW(64,24)
      DIMENSION DFDR(24),DRS(24),DFRR(24),DS(24),DRW(64),DZW(64)
      DIMENSION ALPJS(24),ENU(24,24),GJ(24),TKJ(24),RMX(24,24)
      DIMENSION AMU(24),TALSK(24),COMS(24),RW(64),ZW(64),EKS(24),TK(24)
      DIMENSION ELK(24),EMK(24),ZK(24),DELIJ(24,24),IROW(25),ICOL(25)
      DIMENSION SIGK(24)
      DIMENSION VNB(24),VNF(24),VNW(40),XK(24)
      EQUIVALENCE (DFDR(1),C(1)),(DRS(1),C(25)),(DFRR(1),C(49)),
     1 (DS(1),C(73)),(DRW(1),C(97)),(DZW(1),C(161)),(ALPJS(1),C(225)),
     2 (ENU(1),C(249)),(GJ(1),C(825)),(TKJ(1),C(849)),(RMX(1),C(873)),
     3 (AMU(1),C(1449)),(TALSK(1),C(1473)),(COMS(1),C(1497)),
     4 (RW(1),C(1521)),(ZW(1),C(1585)),(EKS(1),C(1649)),(TK(1),C(1673)),
     5 (ELK(1),C(1697)),(DELIJ(1),C(1721)),(IROW(1),C(2297)),
     6 (ICOL(1),C(2322)),(SIGK(1),C(2347)),(EMK(1),C(2371)),
     7 (ZK(1),C(2395)),(XK(1),C(2419))
      KUN=6
      LUN=7
      NDIM=24
      RHO=0.05
      BETAC=0.5
      WO=.01
      PI=3.14159267
      READ 94,ITITLE
   94 FORMAT (12A6)
      READ 95,M,N,ITA,IRS,WA,WB,EPS,RLGTH,BOND
   95 FORMAT (4I5,5F10.0)
      PRINT 96,ITITLE,M,N,ITA,WA,WB,EPS,IRS
   96 FORMAT (26H1TRIANGULAR MESH GENERATOR/,1X,12A6/,
     117H LOGICAL MESH IS ,I5,3H X ,I5/,32H MAXIMUM NUMBER OF ITERATIONS
     2 IS,I5/,40H RELAXATION FOR LINEAR APPROXIMATION IS ,F7.3/,
     353H INITIAL RELAXATION FACTOR FOR NONLINEAR SOLUTION IS ,F7.3/,
     428H EPSILON FOR CONVERGENCE IS ,F10.6/,19H RESTART IS NUMBER ,I5/)
      NM1=N-1
      MM1=M-1
      NM2=N-2
      EM=1.0
      MM=1
      NPMM1=N+M-1
      IF (MM .EQ. 0) NM2=NM1
      DO 97 J=2,NM1
      DO 97 I=2,MM1
      C(I,J,1)=0.5
      C(I,J,2)=0.5
      C(I,J,3)=0.5
   97 CONTINUE
      IF (IRS .EQ. 0) GO TO 3003
      GO TO (3001,1061,3006,3008,3012,377,389),IRS
 3003 CONTINUE
      READ 200,(X(1,J),Y(1,J),J=1,N)
      READ 200,(X(I,N),Y(I,N),I=2,MM1)
      READ 200,(X(M,J),Y(M,J),J=1,N)
      READ 200,(X(I,1),Y(I,1),I=2,MM1)
  200 FORMAT (8F10.0)
      DO 93 J=1,N
      X(1,J)=X(1,J)/RLGTH
```

```
      Y(1,J)=Y(1,J)/RLGTH
      X(M,J)=X(M,J)/RLGTH
   93 Y(M,J)=Y(M,J)/RLGTH
      DO 92 I=2,MM1
      X(I,N)=X(I,N)/RLGTH
      Y(I,N)=Y(I,N)/RLGTH
      X(I,1)=X(I,1)/RLGTH
   92 Y(I,1)=Y(I,1)/RLGTH
C     INTERPOLATE FROM BOUNDARIES TO GET FIRST GUESS
      DO 100 J=2,NM1
      DO 100 I=2,MM1
      X(I,J)=X(I,J-1)+X(I-1,J)-X(I-1,J-1)
      Y(I,J)=Y(I,J-1)+Y(I-1,J)-Y(I-1,J-1)
  100 CONTINUE
      GO TO 3002
C     RESTART 1
 3001 READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
      WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
 3002 CONTINUE
      IDIR=-1
      DO 103MOST=1,ITA
      SX=0.0
      KIT=MOST
      SY=0.0
      SRX=0.0
      SRY=0.0
      IDIR=-IDIR
      DO 102 K=2,MM1
      DO 102 L=2,NM1
      IF(IDIR) 1001,1002,1002
 1001 I=MM1-K+2
      J=NM1-L+2
      GO TO 1003
 1002 I=K
      J=L
 1003 CONTINUE
      XTEMP=(X(I-1,J-1)+X(I+1,J+1)+X(I,J-1)+X(I,J+1)+X(I-1,J)+X(I+1,J)
     1 )/6.0
      YTEMP=(Y(I-1,J-1)+Y(I+1,J+1)+Y(I,J-1)+Y(I,J+1)+Y(I-1,J)+Y(I+1,J)
     1 )/6.0
      RX=(X(I,J)-XTEMP)*WA
      RY=(Y(I,J)-YTEMP)*WA
      X(I,J)=X(I,J)-RX
      Y(I,J)=Y(I,J)-RY
      SX=SX+X(I,J)**2
      SY=SY+Y(I,J)**2
      SRX=SRX+RX**2
      SRY=SRY+RY**2
  102 CONTINUE
      IF (MOST-20*(MOST/20)) 1022,1021,1022
 1021 WRITE (KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
      REWIND KUN
      IRS=1
      IF(MOST .EQ. 20) PRINT 410,IRS
  410 FORMAT(21H0YOU MAY USE RESTART ,I4)
 1022 CONTINUE
      SRX=SQRT(SRX/SX)
      SRY=SQRT(SRY/SY)
      IF (SRX .LE. EPS .AND. SRY .LE. EPS) 105,103
  103 CONTINUE
      PRINT 201,SRX,SRY
  201 FORMAT(51H1PROCESS DID NOT CONVERGE FOR INITIAL APPROXIMATION/,
     110H EPS-X IS ,E15.7,12H   EPS-Y IS ,E15.7//,41H X AND Y VALUES ARE
     2 PRINTED BELOW BY ROWS/)
      DO 104 I=1,M
  104 PRINT 202,(X(I,J),Y(I,J),J=1,N)
```

```
  202 FORMAT (8E15.7)
      WRITE (KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
      REWIND KUN
      IRS=1
      PRINT 410,IRS
      STOP
  105 PRINT 203,KIT
  203 FORMAT (58H1PROCESS CONVERGED FOR INITIAL APPROXIMATION ON ITERATI
     1ON ,I5/,33H X AND Y ARE PRINTED BELOW BY ROW/)
      DO 106 I=1,M
      PRINT 202,(X(I,J),Y(I,J),J=1,N)
  106 CONTINUE
      WRITE (KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      REWIND KUN
      IRS=2
      PRINT 410,IRS
      GO TO 1062
C     RESTART 2
 1061 READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
 1062 CONTINUE
      WX=WB
      WY=WB
      IDIR=-1
      DO 135 MOST=1,ITA
      IDIR=-IDIR
      KIT=MOST
      SRX=0.0
      SRY=0.0
      SX=0.0
      SY=0.0
      DO 110 I=2,MM1
      DO 110 J=2,NM1
      DXDX=((X(I-1,J-1)+2.0*X(I-1,J)+X(I,J+1))-(X(I,J-1)+2.0*X(I+1,J)
     1 +X(I+1,J+1)))/6.0
      DYDX=((Y(I-1,J-1)+2.0*Y(I-1,J)+Y(I,J+1))-(Y(I,J-1)+2.0*Y(I+1,J)
     1 +Y(I+1,J+1)))/6.0
      GAMMA=DXDX**2+DYDX**2
      DXDY=((X(I-1,J)+2.0*X(I,J+1)+X(I+1,J+1))-(X(I-1,J-1)+2.0*X(I,J-1)
     1 +X(I+1,J)))/6.0
      DYDY=((Y(I-1,J)+2.0*Y(I,J+1)+Y(I+1,J+1))-(Y(I-1,J-1)+2.0*Y(I,J-1)
     1 +Y(I+1,J)))/6.0
      ALPHA=DXDY**2+DYDY**2
      BETA=DXDY*DXDX+DYDX*DYDY
      CP1=ALPHA-BETA
      CP2=BETA
      CP3=GAMMA-BETA
      CP1=BETAC*CP1+(1.0-BETAC)*C(I,J,1)
      CP2=BETAC*CP2+(1.0-BETAC)*C(I,J,2)
      CP3=BETAC*CP3+(1.0-BETAC)*C(I,J,3)
      C(I,J,1)=CP1
      C(I,J,2)=CP2
      C(I,J,3)=CP3
  110 CONTINUE
      DO 120 K=2,MM1
      DO 120 L=2,NM1
      IF(IDIR) 1011,1012,1012
 1011 I=MM1-K+2
      J=NM1-L+2
      GO TO 1013
 1012 I=K
      J=L
 1013 CONTINUE
```

```
      C1=C(I,J,1)
      C2=C(I,J,2)
      C3=C(I,J,3)
      C4=C1
      C5=C2
      C6=C3
      SUMC=C1+C2+C3+C4+C5+C6
      IF (ABS(SUMC) - 1.0E-10) 120,120,111
  111 XTEMP=(C1*X(I-1,J)+C2*X(I-1,J-1)+C3*X(I,J-1)+C4*X(I+1,J)
     1 +C5*X(I+1,J+1)+C6*X(I,J+1))/SUMC
      YTEMP=(C1*Y(I-1,J)+C2*Y(I-1,J-1)+C3*Y(I,J-1)+C4*Y(I+1,J)
     1 +C5*Y(I+1,J+1)+C6*Y(I,J+1))/SUMC
      RX=(X(I,J)-XTEMP)*WX
      RY=(Y(I,J)-YTEMP)*WY
      X(I,J)=X(I,J)-RX
      Y(I,J)=Y(I,J)-RY
      SRX=SRX+RX**2
      SRY=SRY+RY**2
      SX=SX+X(I,J)**2
      SY=SY+Y(I,J)**2
  120 CONTINUE
      IF (MOST-20*(MOST/20)) 3005,3004,3005
 3004 WRITE (KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      REWIND KUN
      IRS=2
      IF(MOST .EQ. 20) PRINT 410,IRS
 3005 CONTINUE
      IF (KIT-1) 126,126,127
  126 SPRX=SRX
      SPRY=SRY
  127 EPSCX=SQRT(SRX/SX)
      EPSCY=SQRT(SRY/SY)
      IF (EPSCX .LE. EPS .AND. EPSCY .LE. EPS) GO TO 140
      ETAX=SQRT(SRX/SPRX)
      ETAY=SQRT(SRY/SPRY)
      ELX=(WX+ETAX-1.0)/(WX*SQRT(ETAX))
      ELY=(WY+ETAY-1.0)/(WY*SQRT(ETAY))
      IF (ABS(ELX)-1.0) 129,129,128
  128 WAX=WX
      GO TO 130
  129 WAX=2.0/(1.0+SQRT(1.0-ELX**2))-WO
  130 IF (ABS(ELY)-1.0) 132,132,131
  131 WAY=WY
      GO TO 133
  132 WAY=2.0/(1.0+SQRT(1.0-ELY**2))-WO
  133 WX=RHO*WAX+(1.0-RHO)*WX
      WY=RHO*WAY+(1.0-RHO)*WY
      SPRX=SRX
      SPRY=SRY
  135 CONTINUE
      PRINT 204,KIT,EPSCX,EPSCY
  204 FORMAT (43H1NONLINEAR SOLUTION DID NOT CONVERGE AFTER ,I5,11H ITER
     1ATIONS/,10H EPS-X IS ,E15.7,12H   EPS-Y IS ,E15.7/,34H X AND Y ARE
     2 PRINTED BELOW BY ROWS/)
      DO 136 I=1,M
      PRINT 202,(X(I,J),Y(I,J),J=1,N)
  136 CONTINUE
      WRITE (KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      IRS=2
      PRINT 410,IRS
      STOP
  140 PRINT 205,KIT
  205 FORMAT (42H1NONLINEAR SOLUTION CONVERGED ON ITERATION,I5/,
     134H X AND Y ARE PRINTED BELOW BY ROWS/)
```

```
      DO 141 I=1,M
      PRINT 202,(X(I,J),Y(I,J),J=1,N)
  141 CONTINUE
      WRITE (KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      IRS=3
      PRINT 410,IRS
      GO TO 3007
C     RESTART 3
 3006 READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
 3007 CONTINUE
      DO 301 J=1,N
      DO 301 I=1,M
      DO 300 K=1,6
      C(I,J,K)=1.0
  300 CONTINUE
      C(I,J,7)=0.0
  301 CONTINUE
      DO 302 J=1,N
      C(1,J,1)=0.0
      C(1,J,2)=0.0
      C(1,J,6)=0.0
      C(M,J,3)=0.0
      C(M,J,4)=0.0
  302 C(M,J,5)=0.0
      DO 303 I=1,M
      C(I,1,1)=0.0
      C(I,1,2)=0.0
      C(I,1,3)=0.0
      C(I,N,4)=0.0
      C(I,N,5)=0.0
  303 C(I,N,6)=0.0
      DO 350 I=1,M
      DO 350 J=1,N
      DO 305 K=1,6
      A(K)=0.0
      R(K)=0.0
      SS(K)=1.0
      T(K)=1.0
      CT(K)=0.0
      CS(K)=0.0
      ELAMDA(K)=C(I,J,K)
      XT(K)=0.0
  305 YT(K)=0.0
      XTEMP=X(I,J)
      YTEMP=Y(I,J)
      IF (I-1) 307,307,306
  306 XT(1)=X(I-1,J)
      YT(1)=Y(I-1,J)
  307 IF (I-M) 308,309,309
  308 XT(4)=X(I+1,J)
      YT(4)=Y(I+1,J)
  309 IF (J-1) 311,311,310
  310 XT(3)=X(I,J-1)
      YT(3)=Y(I,J-1)
  311 IF (J-N) 312,313,313
  312 XT(6)=X(I,J+1)
      YT(6)=Y(I,J+1)
  313 IF (I-1) 316,316,314
  314 IF (J-1) 316,316,315
  315 XT(2)=X(I-1,J-1)
      YT(2)=Y(I-1,J-1)
  316 IF (I-M) 317,319,319
```

```
317 IF (J-N) 318,319,319
318 XT(5)=X(I+1,J+1)
    YT(5)=Y(I+1,J+1)
319 CONTINUE
    XT(7)=XT(1)
    YT(7)=YT(1)
    DO 323 K=1,6
320 SS(K)=SQRT((XT(K)-XTEMP)**2+(YT(K)-YTEMP)**2)
    IF (SS(K) .EQ. 0) SS(K)=1.0
    IF (K-6) 321,322,322
321 T(K)=SQRT((XT(K+1)-XT(K))**2+(YT(K+1)-YT(K))**2)
    IF (T(K) .EQ. 0) T(K)=1.0
    R(K)=(XTEMP+XT(K)+XT(K+1))/3.0
    GO TO 323
322 T(K)=SQRT((XT(6)-XT(1))**2+(YT(6)-YT(1))**2)
    IF (T(K) .EQ. 0) T(K)=1.0
    R(K)=(XTEMP+XT(6)+XT(1))/3.0
323 CONTINUE
    DO 340 K=1,6
    IF(K-1) 324,324,326
324 IF(ELAMDA(6)) 325,331,325
325 COSS=(T(6)**2+SS(6)**2-SS(1)**2)/(2.0*T(6)*SS(6))
    GO TO 328
326 IF(ELAMDA(K-1)) 327,331,327
327 COSS=(T(K-1)**2+SS(K-1)**2-SS(K)**2)/(2.0*T(K-1)*SS(K-1))
328 CS(K)=1.0-COSS**2
    IF(CS(K)) 329,329,330
329 CS(K)=0.0001
330 CS(K)=COSS/SQRT(CS(K))
331 IF(ELAMDA(K)) 332,340,332
332 XA=0.5*(XT(K)+XTEMP)
    XB=(XT(K)+XT(K+1)+XTEMP)/3.0
    XC=0.5*(XT(K+1)+XTEMP)
    YA=0.5*(YT(K)+YTEMP)
    YB=(YT(K)+YT(K+1)+YTEMP)/3.0
    YC=0.5*(YT(K+1)+YTEMP)
    AB=SQRT((XA-XB)**2+(YA-YB)**2)
    BC=SQRT((YB-YC)**2+(XB-XC)**2)
    AD=SQRT((XA-XTEMP)**2+(YA-YTEMP)**2)
    BD=SQRT((XB-XTEMP)**2+(YB-YTEMP)**2)
    CD=SQRT((XC-XTEMP)**2+(YC-YTEMP)**2)
    ARA=0.5*(BC+CD+BD)
    ARA=SQRT(ARA*(ARA-BC)*(ARA-CD)*(ARA-BD))
    ARB=0.5*(AB+BD+AD)
    ARB=SQRT(ARB*(ARB-AB)*(ARB-BD)*(ARB-AD))
    A(K)=ARA+ARB
    IF(K-6) 334,333,333
333 COST=(T(6)**2+SS(1)**2-SS(6)**2)/(2.0*T(6)*SS(1))
    GO TO 335
334 COST=(T(K)**2+SS(K+1)**2-SS(K)**2)/(2.0*T(K)*SS(K+1))
335 CT(K)=1.0-COST**2
    IF (CT(K)) 336,336,337
336 CT(K)=0.0001
337 CT(K)=   COST/SQRT(CT(K))
340 CONTINUE
    K=1
    IF (I-1) 341,341,342
341 IF ( J .EQ. 1) SS(3)=0.0
    IF ( J .EQ. N) SS(6)=0.0
    CV(J)=(.5*(SS(3)+SS(6)))*XTEMP
342 DO 343 K=2,6
    W(K)=ELAMDA(K)*R(K)*CT(K)+ELAMDA(K-1)*R(K-1)*CS(K)
343 W(K)=0.5*W(K)
    W(1)=ELAMDA(1)*R(1)*CT(1)+ELAMDA(6)*R(6)*CS(1)
    W(1)=0.5*W(1)
    IF (XTEMP) 345,344,345
```

```
 344 EMR=0.0
     GO TO 346
 345 EMR=EM*EM/XTEMP
 346 SUMC=0.0
     DO 347 K=1,6
     C(I,J,K)=W(K)
 347 SUMC=SUMC+W(K)+A(K)*EMR
     C(I,J,8)=SUMC
     IF (J-1) 348,348,350
 348 IF (MM) 349,350,349
 349 C(I,J,1)=0.0
     C(I,J,2)=0.0
     C(I,J,3)=0.0
     C(I,J,4)=0.0
     C(I,J,5)=0.0
     C(I,J,6)=0.0
     C(I,J,8)=1.0
 350 CONTINUE
C     SET COEFFICIENTS FOR CONTACT POINT
     C(1,N,1)=0.0
     C(1,N,2)=0.0
     C(1,N,5)=0.0
     C(1,N,6)=0.0
     READ 200,GFLG,GAMMA,THETAC
     THETAC=.0174532925*THETAC
     R(1)=X(1,N-2)
     R(2)=X(1,N-1)
     R(3)=X(1,N)
     YT(1)=Y(1,N-2)
     YT(2)=Y(1,N-1)
     YT(3)=Y(1,N)
     CALL SFIT(FR,FRR,XTEMP,YTEMP)
C     IF GFLG IS 0 COMPUTE GAMMA
     IF (GFLG) 3508,3503,3508
3503 R(3)=X(1,N)
     R(2)=X(2,N)
     R(1)=X(3,N)
     YT(3)=Y(1,N)
     YT(2)=Y(2,N)
     YT(1)=Y(3,N)
     CALL SFIT(ZR,ZRR,XTEMP,YTEMP)
     ZTEMP=COS(THETAC)
     IF(ABS(ZTEMP).GE.1.0) GO TO 3506
     GAMMA=(ABS(ZRR)-ABS(FRR)*ZTEMP)/SIN(THETAC)
     GO TO 3508
3506 GAMMA=0.0
3508 CONTINUE
3509 CONTINUE
     PRINT 4000,GAMMA
4000 FORMAT(7H0GAMMA ,E15.8)
     WRITE (KUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
     II=1
     IRS=4
     PRINT 410,IRS
     NM1=N
     GO TO 3011
C     RESTART 4
3008 II=1
     NM1=N
     READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
    1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
     WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
    1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
     READ (LUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8).(CV(J),J=1,N)
     WRITE(KUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
3009 READ (LUN) (F(J,II),J=1,NM1)
```

```
      IF (EOF,LUN) 3011,3010
 3010 II=II+1
      WRITE(KUN) (F(J,II),J=1,NM1)
      GO TO 3009
 3011 CONTINUE
      DO 355 J=1,N
      DO 355 I=1,M
  355 PHI(I,J)=0.0
      IF (MM .NE. 0) CV(1)=0.0
      DO 360 ICOUNT=II,NM1
      C(1,ICOUNT,7)=CV(ICOUNT)
      CALL SOREL(1)
      DO 356 J=1,NM1
      F(J,ICOUNT)=PHI(1,J)
  356 CONTINUE
      WRITE (KUN) (F(J,ICOUNT),J=1,NM1)
      C(1,ICOUNT,7)=0.0
  360 CONTINUE
      IRS=5
      NM1=N
      PRINT 410,IRS
      GO TO 3013
C     RESTART 5
 3012 READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      NM1=N
      WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      READ (LUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
      WRITE(KUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
      DO 3014 I=1,NM1
 3014 READ (LUN) (F(J,I),J=1,NM1)
      WRITE(KUN) (F(J,I),J=1,NM1)
 3013 CONTINUE
      DO 361 I=1,NM1
      DO 361 J=1,NM1
      EVAL(I,J)=F(I,J)
  361 CONTINUE
      CALL EIGEN(NM1,EVAL,EVI,EVEC,50,3,0,1,NDIM)
      DO 372 I=1,NM1
      NVAL(I)=I
      IF (EVAL(I,I)) 370,370,371
  370 EIGVAL(I)=10.0E+60
      GO TO 372
  371 EIGVAL(I)=1.0/EVAL(I,I)
  372 CONTINUE
      DO 375 I=1,NM1
      K=I
      DO 374 J=K,NM1
      IF (EIGVAL(I)-EIGVAL(J)) 374,374,373
  373 XTEMP=EIGVAL(J)
      II=NVAL(J)
      EIGVAL(J)=EIGVAL(I)
      NVAL(J)=NVAL(I)
      EIGVAL(I)=XTEMP
      NVAL(I)=II
  374 CONTINUE
  375 CONTINUE
      PRINT 400,ITITLE
  400 FORMAT (1H1,12A6/)
      PRINT 401
  401 FORMAT (77H EIGENVALUES AND EIGENVECTORS ASSOCIATED WITH INFLUENCE
     1 COEFFICIENT SOLUTIONS)
      NM2=N
      DO 376 I=1,NM2
      PRINT 402,I,EIGVAL(I)
```

```
  402 FORMAT (6HOMODE ,I3,14H   EIGENVALUE ,E15.8)
      K=NVAL(I)
      PRINT 202,(EVEC(J,K),J=1,NM1)
  376 CONTINUE
      WRITE (KUN) (EIGVAL(I),I=1,NM1),(NVAL(I),I=1,NM1),
     1((EVEC(I,J),I=1,NM1),J=1,NM1),(NVAL(J),J=1,NM1)
      II=1
      IRS=6
      PRINT 410,IRS
      NM1=N
      NM2=N-1
      GO TO 382
C     RESTART 6
  377 READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      NM1=N
      NM2=N-1
      WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      READ (LUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
      WRITE(KUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
      DO 378 I=1,NM1
      READ (LUN) (F(J,I),J=1,NM1)
  378 WRITE(KUN) (F(J,I),J=1,NM1)
      READ  (LUN) (EIGVAL(I),I=1,NM1),(NVAL(I),I=1,NM1),
     1((EVEC(I,J),I=1,NM1),J=1,NM1),(NVAL(J),J=1,NM1)
      WRITE (KUN) (EIGVAL(I),I=1,NM1),(NVAL(I),I=1,NM1),
     1((EVEC(I,J),I=1,NM1),J=1,NM1),(NVAL(J),J=1,NM1)
  379 READ (LUN)(PHIS(I,II),I=1,N),(PHIW(I,II),I=1,NPMM1)
      IF (EOF,LUN) 382,381
  381 WRITE(KUN)(PHIS(I,II),I=1,N),(PHIW(I,II),I=1,NPMM1)
      II=II+1
      GO TO 379
  382 CONTINUE
      DO 383 J=1,N
      DO 383 I=2,M
  383 PHI(I,J)=0.0
      DO 388 ICOUNT=II,NM2
      L=NVAL(ICOUNT)
      DO 384 I=1,NM1
  384 PHI(1,I)=EVEC(I,L)
      CALL SOREL(2)
      DO 385 J=1,N
  385 PHIS(J,ICOUNT)=PHI(1,J)
      DO 386 J=1,N
  386 PHIW(J,ICOUNT)=PHI(M,J)
      DO 387 K=1,MM1
      KPN=K+N
      L=M-K
  387 PHIW(KPN,ICOUNT)=PHI(L,N)
      WRITE (KUN) (PHIS(I,ICOUNT),I=1,N),(PHIW(I,ICOUNT),I=1,NPMM1)
  388 CONTINUE
      NM1=N
      NM2=N-1
      GO TO 391
C     RESTART 7
  389 READ (LUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      NM1=N
      NM2=N-1
      WRITE(KUN) ((X(I,J),Y(I,J),I=1,M),J=1,N)
     1 ,(((C(I,J,K),I=1,M),J=1,N),K=1,3)
      READ (LUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
      WRITE(KUN) GAMMA,(((C(I,J,K),I=1,M),J=1,N),K=1,8),(CV(J),J=1,N)
      DO 390 I=1,NM1
      READ (LUN) (F(J,I),J=1,NM1)
```

```
  390 WRITE(KUN) (F(J,I),J=1,NM1)
      READ  (LUN) (EIGVAL(I),I=1,NM1),(NVAL(I),I=1,NM1),
     1((EVEC(I,J),I=1,NM1),J=1,NM1),(NVAL(J),J=1,NM1)
      WRITE (KUN) (EIGVAL(I),I=1,NM1),(NVAL(I),I=1,NM1),
     1((EVEC(I,J),I=1,NM1),J=1,NM1),(NVAL(J),J=1,NM1)
      DO 392 J=1,NM2
      READ (LUN) (PHIS(I,J),I=1,N),(PHIW(I,J),I=1,NPMM1)
  392 WRITE(KUN) (PHIS(I,J),I=1,N),(PHIW(I,J),I=1,NPMM1)
  391 CONTINUE
      IRS=7
      PRINT 410,IRS
      NM1=N-1
      NM2=N-2
C     READ OTHER INPUTS
      READ 4002,ZCG,VOL,DEN,RHOU
 4002 FORMAT (5E15.8)
      PRINT 400,ITITLE
      PRINT 4003,BOND,ZCG,VOL,DEN,RHOU
 4003 FORMAT(13H0BOND NUMBER ,E15.8, 7H   ZCG ,E15.8, 9H  VOLUME ,E15.8
     1 //,22H  LOWER FLUID DENSITY ,E15.8 ,25H DENSITY OF ULLAGE FLUID
     2 ,E15.8)
      ZCG=ZCG/RLGTH
      VOL=VOL/RLGTH**3
      MM1=M-1
      NM1=N-1
      DO 4110 J=2,NM1
      XTEMP=X(1,J+1)-X(1,J)
      IF(XTEMP.EQ.0.) GO TO 4108
      DZDR=(Y(1,J+1)-Y(1,J))/XTEMP
      IF(DZDR)4104,4102,4104
 4102 SIGN=1.0
      GO TO 4106
 4104 SIGN=DZDR/ABSF(DZDR)
 4106 DZDNF=SIGN/SQRT(1.0+DZDR**2)
      DRDNF=-DZDR*DZDNF
      GO TO 4109
 4108 DZDNF=0.0
      DRDNF=-1.0
 4109 VNF(J)=Y(1,J)*DRDNF-X(1,J)*DZDNF
 4110 CONTINUE
      VNF(1)=0.0
      VNF(N)=Y(1,N)*DRDNF-X(1,N)*DZDNF
      MP1=M+1
      MP2=M+2
      DO 4120 I=2,M
      XTEMP=X(MP2-I,N)-X(MP1-I,N)
      IF(XTEMP.EQ.0.) GO TO 4118
      DZDR=(Y(MP2-I,N)-Y(MP1-I,N))/XTEMP
      IF(DZDR)4114,4112,4114
 4112 SIGN=-1.0
      GO TO 4116
 4114 SIGN=DZDR/ABS(DZDR)
 4116 DZDNW=-SIGN/SQRT(1.0+DZDR**2)
      DRDNW=-DZDR*DZDNW
      GO TO 4119
 4118 DZDNW=0.0
      DRDNW=1.0
 4119 VNW(MP2-I)=Y(MP2-I,N)*DRDNW-X(MP2-I,N)*DZDNW
 4120 CONTINUE
      VNW(1)=Y(1,N)*DRDNW-X(1,N)*DZDNW
      DO 4130 J=1,NM1
      XTEMP=X(M,J+1)-X(M,J)
      IF(XTEMP.EQ.0.) GO TO 4128
      DZDR=(Y(M,J+1)-Y(M,J))/XTEMP
      IF(DZDR)4124,4122,4124
 4122 SIGN=1.0
```

```
      GO TO 4126
 4124 SIGN=DZDR/ABS(DZDR)
 4126 DZDNW=-SIGN*DZDR/SQRT(1.0+DZDR**2)
      DRDNW=-DZDR*DZDNW
      GO TO 4129
 4128 DZDNW=0.0
      DRDNW=1.0
 4129 VNB(J)=Y(M,J)*DRDNW-X(M,J)*DZDNW
 4130 CONTINUE
      VNB(N)=Y(M,N)*DRDNW-X(M,N)*DZDNW
      C(1,1,7)=0.0
      SS(3)=SQRT((X(1,N)-X(1,N-1))**2+(Y(1,N)-Y(1,N-1))**2)
      SS(4)=SQRT((X(2,N)-X(1,N))**2+(Y(2,N)-Y(1,N))**2)
      C(1,N,7)=.5*(VNF(N)+VNF(N-1))*.5*SS(3)*X(1,N)+
     1         .5*(VNW(1)+VNW(2))*.5*SS(4)*X(1,N)
      DO 4132 J=2,NM1
      SS(3)=SQRT((X(1,J)-X(1,J-1))**2+(Y(1,J)-Y(1,J-1))**2)
      SS(6)=SQRT((X(1,J+1)-X(1,J))**2+(Y(1,J+1)-Y(1,J))**2)
      C(1,J,7)=VNF(J)*.5*(SS(3)+SS(6))*X(1,J)
 4132 CONTINUE
      DO 4134 I=2,MM1
      SS(1)=SQRT((X(I,N)-X(I-1,N))**2+(Y(I,N)-Y(I-1,N))**2)
      SS(4)=SQRT((X(I+1,N)-X(I,N))**2+(Y(I+1,N)-Y(I,N))**2)
      C(I,N,7)=VNW(I)*.5*(SS(1)+SS(4))*X(I,N)
 4134 CONTINUE
      SS(1)=SQRT((X(M,N)-X(M-1,N))**2+(Y(M,N)-Y(M-1,N))**2)
      SS(3)=SQRT((X(M,N)-X(M,N-1))**2+(Y(M,N)-Y(M,N-1))**2)
      C(M,N,7)=(.5*(VNW(M)+VNW(M-1))*.5*SS(3)
     1         +.5*(VNB(N)+VNB(N-1))*.5*SS(1))*X(M,N)
      DO 4136 J=2,NM1
      SS(3)=SQRT((X(M,J)-X(M,J-1))**2+(Y(M,J)-Y(M,J-1))**2)
      SS(6)=SQRT((X(M,J+1)-X(M,J))**2+(Y(M,J+1)-Y(M,J))**2)
      C(M,J,7)=VNB(J)*.5*(SS(3)+SS(6))*X(M,J)
 4136 CONTINUE
      SS(6)=SQRT((X(M,2)-X(M,1))**2+(Y(M,2)-Y(M,1))**2)
      C(M,1,7)=VNB(1)*.5*SS(6)*.25*(3.*X(M,1)+X(M,2))
      DO 4138 I=1,M
      DO 4138 J=1,N
      PHI(I,J)=0.0
 4138 CONTINUE
      CALL SOREL(1)
      XTEMP=0.0
      DO 4140 J=1,NM1
      XTEMP=XTEMP+.5*(PHI(1,J)*VNF(J)*X(1,J)
     1         +PHI(1,J+1)*VNF(J+1)*X(1,J+1))*
     2         SQRT((Y(1,J+1)-Y(1,J))**2+(X(1,J+1)-X(1,J))**2)
 4140 CONTINUE
      DO 4142 I=1,MM1
      XTEMP=XTEMP+.5*(PHI(I,N)*VNW(I)*X(I,N)
     1         +PHI(I+1,N)*VNW(I+1)*X(I+1,N))*
     2         SQRT((Y(I+1,N)-Y(I,N))**2+(X(I+1,N)-X(I,N))**2)
 4142 CONTINUE
      DO 4144 J=1,NM1
      XTEMP=XTEMP+.5*(PHI(M,J)*VNB(J)*X(M,J)
     1         +PHI(M,J+1)*VNB(J+1)*X(M,J+1))*
     2         SQRT((Y(M,J+1)-Y(M,J))**2+(X(M,J+1)-X(M,J))**2)
 4144 CONTINUE
      SIF=XTEMP*PI/VOL
      DO 415 J=1,NM1
      DRS(J)=X(1,J+1)-X(1,J)
  415 CONTINUE
      DRS(N)=DRS(NM1)
      DFDR(1)=0.0
      DFRR(1)=2.0*(Y(1,2)-Y(1,1))/DRS(1)**2
      DO 417 J=2,NM1
      R(1)=X(1,J-1)
```

```
      R(2)=X(1,J)
      R(3)=X(1,J+1)
      YT(1)=Y(1,J-1)
      YT(2)=Y(1,J)
      YT(3)=Y(1,J+1)
      CALL SFIT(FR,YTEMP,ZTEMP,XTEMP)
      DFDR(J)=XTEMP
      DFRR(J)=YTEMP
      XK(J)=ZTEMP
  417 CONTINUE
      DFDR(N)=FR
      DFRR(N)=DFRR(N-1)
      XK(1)=XK(2)
      XK(N)=YTEMP
      DO 418 J=1,NM1
  418 DS(J)=2.0*PI*(X(1,J+1)+X(1,J))*.5
     1 *SQRT((Y(1,J+1)-Y(1,J))**2+(X(1,J+1)-X(1,J))**2)
      DS(N)=DS(NM1)
      DO 419 J=1,NM1
      DRW(J)=X(M,J+1)-X(M,J)
      RW(J)=X(M,J)
      ZW(J)=Y(M,J)
  419 DZW(J)=Y(M,J+1)-Y(M,J)
      RW(N)=X(M,N)
      ZW(N)=Y(M,N)
      DO 420 K=1,MM1
      ITEMP=M-K+1
      DRW(N+K-1)=X(M-K,N)-X(ITEMP,N)
      DZW(N+K-1)=Y(M-K,N)-Y(ITEMP,N)
      RW(N+K)=X(M-K,N)
  420 ZW(N+K)=Y(M-K,N)
      RW(N+M)=X(1,N)
      ZW(N+M)=Y(1,N)
      DRW(NPMM1)=DRW(N+M-2)
      DZW(NPMM1)=DZW(N+M-2)
      DRW(N+M)=DRW(NPMM1)
      DZW(N+M)=DZW(NPMM1)
      DO 422 J=1,NM2
      XTEMP=0.0
      DO 421 K=1,NM1
  421 XTEMP=XTEMP+0.5*(PHIS(K,J)**2+PHIS(K+1,J)**2)*DS(K)
  422 ALPJS(J)=XTEMP
      DO 423 J=1,NM2
      DO 423 I=1,NM2
      ENU(I,J)=2.0*PI/ALPJS(I)*X(1,N)*PHIS(N,J )*PHIS(N,I)*EIGVAL(J)
  423 CONTINUE
C     READ NUMBER OF EIGENVALUES TO BE USED IN REMAINING CALCULATIONS
      READ 95,NEV
      NM2=NEV
      DO 430 J=1,NM2
      DO 430 I=1,NM2
      EPSIJ=0.0
      BETAIJ=0.0
      DELIJ(I,J)=0.0
      GAMIJ=0.0
      XTEMP=0.0
      YTEMP=0.0
      IF(I.EQ.J) DELIJ(I,J)=1.0
      DO 428 K=1,NM1
      YPRP=(Y(1,K+1)-Y(1,K))/DRS(K)
      EE=DRS(K)/ABS(DRS(K))
      CURVA=1.0+YPRP**2
      CURVB=SQRT(CURVA)
      EPSIJ=EPSIJ+0.5*(PHIS(K,J)*PHIS(K,I)/((X(1,K)+.00001)**2)
     1 +PHIS(K+1,J)*PHIS(K+1,I)/(X(1,K+1)**2))*DS(K)
      DPSIIA=(PHIS(K+1,I)-PHIS(K,I))/DRS(K)
```

```
      DPSIJA=(PHIS(K+1,J)-PHIS(K,J))/DRS(K)
      ZTEMP=YPRP**2/(CURVA*(0.5*(X(1,K)+X(1,K+1)))**2)+
     1      0.5*(XK(K)**2+XK(K+1)**2)
      XTEMP=XTEMP+DPSIIA*DPSIJA*DS(K)/CURVA
      YTEMP=YTEMP-.5*ZTEMP*(PHIS(K,I)*PHIS(K,J)+PHIS(K+1,I)*PHIS(K+1,J))
     1      *DS(K)
      BETAIJ=BETAIJ+0.5*(PHIS(K,I)*PHIS(K,J)+PHIS(K+1,I)*PHIS(K+1,J))
     1       *DS(K)/CURVB*EE
  428 CONTINUE
      EPSIJ=EPSIJ*EIGVAL(J)/ALPJS(I)
      GAMIJ=(XTEMP+YTEMP)*EIGVAL(J)/ALPJS(I)
      BETAIJ=BETAIJ*EIGVAL(J)/ALPJS(I)
  430 RMX(I,J)=-GAMMA*ENU(I,J)+GAMIJ+EM*EM*EPSIJ+BOND*BETAIJ
     1         *(DEN-RHOU)/DEN
      CALL MATINV(DELIJ,IROW,ICOL,NM2,NDIM,1.0E-06)
      DO 432 I=1,NM2
      DO 432 J=1,NM2
      XTEMP=0.0
      DO 431 K=1,NM2
  431 XTEMP=XTEMP+RMX(I,K)*DELIJ(K,J)
  432 EVEC(I,J)=XTEMP
      DO 433 I=1,NM2
      DO 433 J=1,NM2
  433 RMX(I,J)=EVEC(I,J)
      KM2=NM2
      CALL EIGEN(NM2,RMX,EVI,EVEC,50,2,0,1,NDIM)
C     PUT EIGENVALUES AND VECTORS IN INCREASING ORDER
      DO 442 I=1,NM2
      NVAL(I)=I
      IF (RMX(I,I)) 440,440,441
  440 KM2=KM2-1
      COMS(I)=10.0E+60
      GO TO 442
  441 COMS(I)=RMX(I,I)
  442 CONTINUE
      DO 445 I=1,NM2
      K=I
      DO 444 J=K,NM2
      IF (COMS(I)-COMS(J)) 444,444,443
  443 XTEMP=COMS(J)
      II=NVAL(J)
      COMS(J)=COMS(I)
      NVAL(J)=NVAL(I)
      COMS(I)=XTEMP
      NVAL(I)=II
  444 CONTINUE
  445 CONTINUE
      DO 447 J=1,NM2
      K=NVAL(J)
      DO 447 I=1,NM2
      F(I,J)=EVEC(I,K)
  447 CONTINUE
      PRINT 4005
 4005 FORMAT (39H0 EIGENVALUES AND EIGENVECTORS AT LOW G)
      DO 448 I=1,NM2
      PRINT 402,I,COMS(I)
      PRINT 202,(F(J,I),J=1,NM2)
  448 CONTINUE
      XTEMP=DFDR(N)
      RII=X(1,N)
      FII=Y(1,N)
      NPMM2=NPMM1-1
      L=N+M
      RWMAX=RW(1)
      ZWMAX=ZW(1)
      DO 4491 I=2,L
```

```
      IF (RW(I)-RWMAX) 4491,449,449
  449 RWMAX=RW(I)
      ZWMAX=ZW(I)
 4491 CONTINUE
      RFMAX=X(1,1)
      ZFMAX=Y(1,1)
      DO 4493 I=2,N
      IF (X(1,I)-RFMAX) 4493,4492,4492
 4492 RFMAX=X(1,I)
      ZFMAX=Y(1,I)
 4493 CONTINUE
      DO 453 J=1,NM1
      XTEMP=0.0
      DO 451  K=1,NPMM2
      IF(ZW(K)-ZWMAX) 450,450,4501
  450 SIGN=1.0
      GO TO 4502
 4501 SIGN=-1.0
 4502 IF(DRW(K)) 4504,4503,4504
 4503 DZDR=1.0E+07
      GO TO 4505
 4504 DZDR=DZW(K)/DRW(K)
 4505 YTEMP=SQRT(1.0+DZDR**2)
      IF (ABS(DZDR)-1.0E+05) 4506,4507,4507
 4506 DRDN=SIGN*DZDR/YTEMP
      GO TO 4508
 4507 DRDN=1.0
 4508 DZDN=-SIGN/YTEMP
      XT(1)=0.5*(RW(K)+RW(K+1))
      YT(1)=0.5*(ZW(K)+ZW(K+1))
      XT(2)=0.5*(PHIW(K,J)+PHIW(K+1,J))
      YT(2)=SQRT(DRW(K)**2+DZW(K)**2)
      XTEMP=XTEMP+PI*XT(1)*XT(2)*(YT(1)*DRDN-XT(1)*DZDN)*YT(2)
  451 CONTINUE
C     XTEMP NOW CONTAINS MUJ1
      DO 452  K=1,NM1
      IF(DFDR(K))4511,4512,4511
 4511 SIGN=DFDR(K)/ABS(DFDR(K))
      GO TO 4513
 4512 SIGN=1.0
 4513 IF(ABS(DFDR(K))-1.0E+05) 4514,4515,4515
 4514 DRDNA=-SIGN*DFDR(K)/SQRT(1.0+DFDR(K)**2)
      GO TO 4516
 4515 DRDNA=-1.0
 4516 DZDNA=SIGN/SQRT(1.0+DFDR(K)**2)
      IF(DFDR(K+1))4521.4522,4521
 4521 SIGN1=DFDR(K+1)/ABS(DFDR(K+1))
      GO TO 4523
 4522 SIGN1=1.0
 4523 IF(ABS(DFDR(K+1))-1.0E+05)4527,4528,4528
 4527 DRDNB=-SIGN1*DFDR(K+1)/SQRT(1.0+DFDR(K+1)**2)
      GO TO 4529
 4528 DRDNB=-1.0
 4529 DZDNB=SIGN1/SQRT(I.0+DFDR(K+1)**2)
      XT=SQRT(DRS(K)**2+(Y(1,K+1)-Y(1,K))**2)
      XTEMP=XTEMP+0.5*PI*(PHIS(K,J)*(Y(1,K)*DRDNA-X(1,K)*DZDNA)*X(1,K)
     1 +PHIS(K+1,J)*(Y(1,K+1)*DRDNB-X(1,K+1)*DZDNB)*X(1,K+1))*XT
  452 CONTINUE
      AMU(J)=XTEMP
  453 CONTINUE
      IF(NEV .GT. KM2) NEV=KM2
      DO 461 K=1,NEV
      DO 461 I=1,NEV
      TALSK(K)=0.0
      DO 460 J=1,NM1
      XTEMP=0.0
```

```
      YTEMP=0.0
      XT=0.0
      YT=0.0
      DO 459 L=1,NEV
      XTEMP=XTEMP-F(L,K)*(-EIGVAL(L)*PHIS(J,L)*SQRT(1.0+DFDR(J)**2))
      YTEMP=YTEMP+F(L,I)*PHIS(J,L)
      XT=XT-F(L,K)*(-EIGVAL(L)*PHIS(J+1,L)*SQRT(1.0+DFDR(J+1)**2))
  459 YT=YT+F(L,I)*PHIS(J+1,L)
  460 TALSK(K)=(.5*XTEMP*YTEMP*X(1,J)+0.5*XT*YT*X(1,J+1))*ABS(DRS(J))
     1 +TALSK(K)
  461 RMX(I,K)=2.0*PI*TALSK(K)
      CALL MATINV(RMX,IROW,ICOL,NEV,NDIM,1.0E-06)
      DO 464 K=1,NEV
      XTEMP=0.0
      DO 463 J=1,NM1
      XTEMP=XTEMP+0.5*(-EIGVAL(K)*PHIS(J,K)*X(1,J)
     1 -EIGVAL(K)*PHIS(J+1,K)*X(1,J+1))*DS(J)
  463 CONTINUE
  464 GJ(K)=-XTEMP
      DO 466 K=1,NEV
      TALSK(K)=0.0
      DO 465 J=1,NEV
  465 TALSK(K)=TALSK(K)+F(J,K)*GJ(J)
  466 CONTINUE
      DO 4670 I=1,NEV
      XTEMP=0.0
      DO 467 J=1,NEV
  467 XTEMP=XTEMP+RMX(I,J)*TALSK(J)
 4670 EKS(I)=XTEMP
      EMS=1.0
      ZS=0.0
      DO 468 K=1,NEV
  468 SIGK(K)=GJ(K)
      DO 471 K=1,NEV
      XTEMP=0.0
      XT=0.0
      YT=0.0
      DO 470 J=1,NEV
      XTEMP=XTEMP+F(J,K)*SIGK(J)
  470 XT=XT+F(J,K)*AMU(J)
      TK(K)=0.5*XTEMP
      ELK(K)=EKS(K)*XT
      EMK(K)=EKS(K)*TK(K)/VOL
      EMS=EMS-EMK(K)
      ZK(K)=RLGTH/VOL*ELK(K)/EMK(K)
      ZS=ZS+EMK(K)*ZK(K)
  471 CONTINUE
      ZS=1.0/EMS*(ZCG-ZS)
      XTEMP=0.0
      DO 473 K=1,NEV
  473 XTEMP=XTEMP+EMK(K)*ZK(K)**2
      SI0=SIF-(EMS*ZS**2+XTEMP)/RLGTH**2
      EMF=DEN*VOL*RLGTH**3
      PRINT 475,SIF,SI0,EMF
  475 FORMAT(25H0 SIF-MOMENT OF  INERTIA =,E15.8/
     1          25H0 SI0-MOMENT OF  INERTIA =,E15.8/
     2          22H0 EMF-MASS OF LIQUID =,E15.8)
      PRINT 500
  500 FORMAT (7H0 MK/MF/)
      PRINT 202,(EMK(K),K=1,NEV)
      PRINT 501
  501 FORMAT (4H0 ZK/)
      PRINT 202,(ZK(K),K=1,NEV)
      PRINT 502,EMS,ZS
  502 FORMAT(5H0M0= ,E15.8,7H  Z0=  ,E15.8)
      ENDFILE KUN
```

```
      STOP
      END

      SUBROUTINE SFIT(YR,XK3,XK,YR2)
C     FIT QUADRATIC THROUGH THREE POINTS
      COMMON C(40,24,8),PHI(40,24),WB,EPS,ITA,N,M,WO,RHO,R(6),YT(7)
      IF(ABS((YT(3)-YT(2))/(R(3)-R(2))).GT.1.0) GO TO 30
      IF(ABS((YT(2)-YT(1))/(R(2)-R(1))).GT.1.0) GO TO 30
      XTEMP=(R(1)**2-R(3)**2)*(R(2)-R(3))-(R(2)**2-R(3)**2)*(R(1)-R(3))
      YTEMP=(YT(1)-YT(3))*(R(2)-R(3))-(YT(2)-YT(3))*(R(1)-R(3))
      AD=YTEMP/XTEMP
      AB=(YT(2)-YT(3)-AD*(R(2)**2-R(3)**2))/(R(2)-R(3))
      YR=2.0*AD*R(3)+AB
      YR2=2.0*AD*R(2)+AB
      YRR=2.0*AD
      XK=YRR/(1.0+YR2**2)**1.5
      H2=R(2)-R(3)
      H1=R(1)-R(3)
      F2=YT(2)
      F1=YT(1)
      F0=YT(3)
      YRR=2.*(H2*(F1-F0)-H1*(F2-F0))/(H1*H2*(H1-H2))
      XK3=YRR/(1.0+YR**2)**1.5
      RETURN
   30 YTEMP=(YT(1)**2-YT(3)**2)*(YT(2)-YT(3))-(YT(2)**2-YT(3)**2)*
     1      (YT(1)-YT(3))
      XTEMP=(R(1)-R(3))*(YT(2)-YT(3))-(R(2)-R(3))*(YT(1)-YT(3))
      AD=XTEMP/YTEMP
      AB=(R(2)-R(3)-AD*(YT(2)**2-YT(3)**2))/(YT(2)-YT(3))
      YR=1.0/(2.0*AD*YT(3)+AB)
      YR2=1.0/(2.0*AD*YT(2)+AB)
      YRR=-2.0*AD*YR2**3
      XK=YRR/(1.0+YR2**2)**1.5
      H2=YT(2)-YT(3)
      H1=YT(1)-YT(3)
      F2=R(2)
      F1=R(1)
      F0=R(3)
      RYY=2.*(H2*(F1-F0)-H1*(F2-F0))/(H1*H2*(H1-H2))
      YRR3=-RYY*YR**3
      XK3=YRR3/(1.0+YR**2)**1.5
      RETURN
      END
```

```
      SUBROUTINE SOREL(LL)
      COMMON C(40,24,8),PHI(40,24),WB,EPS,ITA,N,M,WO,RHO,R(6),YT(7)
C     SUCCESSIVE OVER-RELAXATION TO SOLVE FOR PHI
      IL=LL
      WX=WB
      DO 135 MOST=1,ITA
      KIT=MOST
      SRX=0.0
      SX=0.0
      DO 120 I=IL,M
      DO 120 J=1,N
      PHIT=C(I,J,7)
      DO 108 K=1,6
      CT=C(I,J,K)
      IF (CT .EQ. 0.0) GO TO 108
      IF (K-2) 100,101,102
  100 II=I-1
      JJ=J
      GO TO 107
  101 II=I-1
      JJ=J-1
      GO TO 107
  102 IF (K-4) 103,104,105
  103 II=I
      JJ=J-1
      GO TO 107
  104 II=I+1
      JJ=J
      GO TO 107
  105 IF (K-5) 109,106,109
  106 II=I+1
      JJ=J+1
      GO TO 107
  109 II=I
      JJ=J+1
  107 PHIT=PHIT+CT*PHI(II,JJ)
  108 CONTINUE
      PHIT=PHIT/C(I,J,8)
      RX=WX*(PHI(I,J)-PHIT)
      PHI(I,J)=PHI(I,J)-RX
      SRX=SRX+RX**2
      SX=SX+PHI(I,J)**2
  120 CONTINUE
      IF (SX .EQ. 0.0) GO TO 140
      IF (KIT-1) 126,126,127
  126 SPRX=SRX
  127 EPSCX=SQRT(SRX/SX)
      IF (EPSCX .LE. EPS) GO TO 140
      ETAX=SQRT(SRX/SPRX)
      ELX=(WX+ETAX-1.0)/(WX*SQRT(ETAX))
      IF (ABS(ELX)-1.0) 129,129,128
  128 WAX=WX
      GO TO 130
  129 WAX=2.0/(1.0+SQRT(1.0-ELX**2))-WO
  130 WX=RHO*WAX+(1.0-RHO)*WX
      SPRX=SRX
  135 CONTINUE
      IF(IL .EQ. 1) PRINT 200,KIT,EPSCX
      IF(IL .EQ. 2) PRINT 204,KIT,EPSCX
  200 FORMAT (*1SOLUTION DID NOT CONVERGE FOR INFLUENCE COEFFICIENTS*/,
     1 * AFTER *,I5,* ITERATIONS  EPSCX= *,E15.7)
  204 FORMAT(*1SOLUTION DID NOT CONVERGE FOR EIGENVECTOR ON SURFACE*/,
     1 * AFTER *,I5,* ITERATIONS   EPSCX= *,E15.8)
```

```
      DO 136 I=1,M
      PRINT 202,(PHI(I,J),J=1,N)
  202 FORMAT (8E15.7)
  136 CONTINUE
      STOP
  140 IF(IL .EQ. 1) PRINT 203,KIT
      IF(IL .EQ. 2) PRINT 205,KIT
  203 FORMAT (* INFLUENCE COEF. SOL. CONVERGED AFTER ITERATION*,I5)
  205 FORMAT(* EIGENVECTOR SOLUTION CONVERGED AFTER ITERATION*,I5)
      RETURN
      END
```

```
      SUBROUTINE EIGEN(N,B,TI,T,MAXIT,NDEC,N1OPT,N2OPT,NDIM)
C     SUBROUTINE FOR GENERATING THE EIGENVALUES AND EIGENVECTORS
C     OF A REAL SYMMETRIC OR NON-SYMMETRIC MATRIX.
C     THIS PROGRAM GENERATES THE EIGENVALUE MATRIX(REAL OR COMPLEX),
C     AND AS OPTIONS, THE EIGENVECTOR MATRIX AND ITS INVERSE.
C       THE CALL FOR THIS SUBROUTINE IS AS FOLLOWS,
C     CALL EIGEN (N,B,TI,T,MAXIT,NDEC,N1OPT,N2OPT,NDIM)
C       WHERE N IS THE ORDER OF THE MATRIX
C             B IS THE MATRIX WHOSE EIGENVALUES ARE DESIRED
C            TI IS THE INVERSE OF THE EIGENVECTOR MATRIX
C             T IS THE EIGENVECTOR MATRIX
C         MAXIT IS MAX NO. OF ITERATIONS FOR GENERATING EIGENVALUES
C        NDEC IS THE NUMBER OF TIMES RESULT IS REFINED
C         N1OPT IS 1 IF OPTION =1 IS DESIRED, OTHERWISE 0.
C     OPTION =1 PROVIDES FOR GENERATING THE EIGENVECTOR MATRIX INVERSE
C         N2OPT IS 1 IF OPTION =2 IS DESIRED, OTHERWISE 0.
C     OPTION =2 PROVIDES FOR GENERATING THE EIGENVECTOR MATRIX
C          NDIM IS DIMENSIONED NO. OF ROWS OF MATRIX (B)
C       THE ORGINAL MATRIX B IS LOST DURING THE COMPUTATIONS AND IS
C     REPLACED BY THE EIGENVALUE MATRIX.
      DIMENSION B(1),TI(1),T(1)
C     INITIALIZE COUNTERS FOR NO. OF ITERATIONS AND YR,YS REDUCTIONS
      IT=0
      NTIMES=0
      ANORM=0.0
      DO 1100 I=1,N
      DO 1100 J=1,N
      IJ=(J-1)*NDIM+I
 1100 ANORM=ANORM+B(IJ)**2
      ANORM=SQRTF(ANORM)
      DO 1101 I=1,N
      DO 1101 J=1,N
      IJ=(J-1)*NDIM+I
 1101 B(IJ)=B(IJ)/ANORM
C     FORM IDENTITY MATRIX IN TI LOCATION IF OPTION 1 IS DESIRED
 1000 IF(N1OPT)1010,1010,1001
 1001 DO 1003 I=1,N
      II = (I-1)*NDIM+I
      DO 1002 J=1,N
      IJ = (J-1)*NDIM+I
 1002 TI(IJ) = 0.
 1003 TI(II) = 1.0
C     FORM IDENTITY MATRIX IN T LOCATION IF OPTION 2 DESIRED
 1010 IF(N2OPT)1020,1020,1011
 1011 DO 1013 I=1,N
      II = (I-1)*NDIM+I
      DO 1012 J=1,N
      IJ = (J-1)*NDIM+I
 1012 T(IJ) = 0.
 1013 T(II) = 1.0
 1020 CONTINUE
      YR=10.0E-7
      YS=10.0E-7
   96 NO=N-1
      SUM=10.0E20
   97 TAU=0.0
      EN=0.0
      DO 1 I=1,NO
      JO=I+1
      DO 1 J=JO,N
      IJ = (J-1)*NDIM+I
      JI = (I-1)*NDIM+J
    1 TAU = TAU+B(IJ)**2+B(JI)**2
```

```
      DO 2 I=1,N
      II = (I-1)*NDIM+I
      TE = B(II)
    2 EN=EN+TE**2
    4 EN=EN+TAU
    5 DELN=SUM-EN
    6 IF(DELN)8,8,7
    7 SUM=EN
      IT=IT+1
      IF(MAXIT-IT)120,120,10
    8 CONTINUE
      IF(NDEC-NTIMES)120,120,9
    9 YR=YR/100.0
      YS=YS/100.0
      NTIMES=NTIMES+1
      IT=IT+1
   10 DO 98 K=1,NO
      KK = (K-1)*NDIM+K
      KO=K+1
   11 DO 98 M=KO,N
      MM = (M-1)*NDIM+M
      KM = (M-1)*NDIM+K
      MK = (K-1)*NDIM+M
      H=0.0
      G=0.0
      HJ=0.0
   12 DO 24 I=1,N
   13 IF(I-K)14,24,14
   14 IF(I-M)15,24,15
   15 IK = (K-1)*NDIM+I
      KI = (I-1)*NDIM+K
      IM = (M-1)*NDIM+I
      MI = (I-1)*NDIM+M
      BO = B(IK)
   16 BR = B(KI)
   17 BQ = B(IM)
   18 BS = B(MI)
   19 H=H+BR*BS-BO*BQ
   20 TEP=BO*BO+BS*BS
   21 TEM=BR*BR+BQ*BQ
   22 G=G+TEP+TEM
   23 HJ=HJ-TEP+TEM
   24 CONTINUE
   25 H=2.0*H
      D = B(KK)-B(MM)
      TEP = B(KM)
      TEM = B(MK)
      C=TEP+TEM
      E=TEP-TEM
   26 IF(ABSF(C)-YR)27,27,30
   27 CC=1.0
   28 SS=0.0
   29 GO TO 39
   30 BY=D/C
   31 IF(BY)400,401,401
  400 SIG=-1.0
      GO TO 32
  401 SIG =1.0
   32 COT=BY+(SIG*SQRTF(BY*BY+1.0))
   33 SS=SIG/SQRTF(COT*COT+1.0)
   34 CC=SS*COT
   35 TEP=CC*CC-SS*SS
   36 TEM=2.0*SS*CC
   37 D=D*TEP+C*TEM
   38 H=H*TEP-HJ*TEM
   39 CONTINUE
```

```
  40 ED=2.0*E*D
  41 EDH=ED-H
  42 DEN=G+2.0*(E*E+D*D)
  43 TEE=EDH/(DEN+DEN)
  75 CONTINUE
 205 IF(ABSF(TEE)-YS)44,44,46
  44 CH=1.0
  45 SH=0.0
     GO TO 48
  46 CH=1.0/SQRTF(1.0-TEE*TEE)
  47 SH=TEE*CH
  48 C1=CH*CC-SH*SS
  49 C2=CH*CC+SH*SS
  50 S1=CH*SS+SH*CC
  51 S2=-CH*SS+SH*CC
  52 CONTINUE
  53 IF(S1)55,54,55
  54 IF(S2)55,98,55
  55 DO 59 J=1,N
     KJ = (J-1)*NDIM+K
     MJ = (J-1)*NDIM+M
  56 BO = B(KJ)
  57 BR = B(MJ)
  58 B(KJ) = C1*BO+S1*BR
  59 B(MJ) = S2*BO+C2*BR
  60 DO 66 J=1,N
     JK = (K-1)*NDIM+J
     JM = (M-1)*NDIM+J
  61 BO = B(JK)
  62 BR = B(JM)
  65 B(JK) = BO*C2-BR*S2
  66 B(JM) = -BO*S1+BR*C1
1070 IF(N1OPT)1075,1075,1071
1071 DO 1072 J=1,N
     KJ = (J-1)*NDIM+K
     MJ = (J-1)*NDIM+M
     BQ = TI(KJ)
     BS = TI(MJ)
     TI(KJ) = C1*BQ+S1*BS
1072 TI(MJ) = S2*BQ+C2*BS
1075 IF(N2OPT)98,98,1076
1076 DO 1077 J=1,N
     JK = (K-1)*NDIM+J
     JM = (M-1)*NDIM+J
     BO = T(JK)
     BR = T(JM)
     T(JK) = BO*C2-BR*S2
1077 T(JM) = -BO*S1+BR*C1
  98 CONTINUE
     GO TO 97
 120 DO 1102 I=1,N
     DO 1102 J=1,N
     IJ=(J-1)*NDIM+I
1102 B(IJ)=B(IJ)*ANORM
     IF (N2OPT) 1107,1107,1103
1103 DO 1106 J=1,N
     ANORM=0.0
     DO 1104 I=1,N
     IJ=(J-1)*NDIM+I
1104 ANORM=ANORM+T(IJ)**2
     ANORM=SQRTF(ANORM)
     DO 1105 I=1,N
     IJ=(J-1)*NDIM+I
1105 T(IJ)=T(IJ)/ANORM
1106 CONTINUE
1107 RETURN
```

```
      SUBROUTINE MPRINT (A,M,N,MD)
      DIMENSION  A(1),IT(6),C(6)
      EQUIVALENCE (IT,C)
    2 FORMAT    (1H0, 4X, 6( 6X, 7HCOLUMN    1I4  )    ///     )
    3 FORMAT    (1H  1I4, X,    (6E 17.8)   )
      N1=N
      N2=6
      N3=6
      N4=1
    4 IF (N3-N1) 6,6,5
    5 N2=N1-N3+6
      N3=N1
    6 K=0
      DO 7  I= N4,N3
        K=K+1
    7   IT(K)=I
      PRINT 2, (IT(I),I=1,N2)
      DO 9  I=1,M
        K=0
        L=MD*(N4-1)+I
        DO 8  J=N4,N3
          K=K+1
      C(K)=A(L)
      L=L+MD
    8 CONTINUE
    9  PRINT 3, I, (C(K),K=1,N2)
      IF (N3-N1) 10,11,11
   10 N3=N3+6
      N4=N4+6
      GO TO 4
   11 RETURN
      END
```

```
      SUBROUTINE MATINV(A,IROW,ICOL,N,NDIM,SMLST)
      DIMENSION A(1),IROW(1),ICOL(1)
C     709-16065
C     709-16065  SUBROUTINE MATINV - MATRIX INVERSION ROUTINE
C
C         A = ARRAY NAME OF MATRIX
C      IROW = DIMENSIONED AT N+1 OR GREATER
C      ICOL = DIMENSIONED AT N OR GREATER
C        N  = NUMBER OF EQUATIONS
C      NDIM = VALUE OF I IN DIMENSION A(I,J) , I AND J MAY DIFFER
C     SMLST = SMALLEST LEADING ELEMENT ALLOWED BEFORE CALLING THE
C             SYSTEM SINGULAR , USUALLY = 1.0 E-04 OR 1.0 E-05
C
      NP1=N+1
      DO 5 I=1,N
      ICOL(I)=I
    5 IROW(I)=I
      DO 75 ITER=1,N
      MAXR=ITER
      MAXC=1
      TEMP=ABSF(A(MAXR))
      LIMITC=NP1-ITER
      DO 15 I=ITER,N
      DO 15 J=1,LIMITC
      IJ=(J-1)*NDIM+I
      IF (TEMP-(ABSF(A(IJ)))) 10,15,15
   10 MAXR=I
      MAXC=J
      TEMP=ABSF(A(IJ))
   15 CONTINUE
      IF (TEMP-SMLST) 20,20,25
   20 IROW(NP1)=ITER
      PRINT 200, ITER,SMLST
  200 FORMAT (7H0ON THEI3,63HTH ITERATION ALL THE REMAINING TERMS WERE L
     1ESS THAN OR EQUAL TO E11.4,18H IN ABSOLUTE VALUE)
      RETURN
   25 IF (MAXR-ITER) 30,40,30
   30 DO 35 J=1,N
      MAXRJ=(J-1)*NDIM+MAXR
      ITJ=(J-1)*NDIM+ITER
      TEMP=A(MAXRJ)
      A(MAXRJ)=A(ITJ)
   35 A(ITJ)=TEMP
      ITEMP=IROW(MAXR)
      IROW(MAXR)=IROW(ITER)
      IROW(ITER)=ITEMP
   40 IF (MAXC-1) 45,55,45
   45 DO 50 I=1,N
      IMAXC=(MAXC-1)*NDIM+I
      TEMP=A(I)
      A(I)=A(IMAXC)
   50 A(IMAXC)=TEMP
      ITEMP=ICOL(MAXC)
      ICOL(MAXC)=ICOL(1)
      ICOL(1)=ITEMP
   55 TEMP=A(ITER)
      ITEMP=ICOL(1)
      DO 60 J=2,N
      ITJM1=(J-2)*NDIM+ITER
      ITJ=(J-1)*NDIM+ITER
      A(ITJM1)=A(ITJ)/TEMP
   60 ICOL(J-1)=ICOL(J)
      ITN=(N-1)*NDIM+ITER
```

```
      A(ITN)=1.0/TEMP                                                   MATIN063
      ICOL(N)=ITEMP                                                     MATIN064
      DO 75 I=1,N                                                       MATIN065
      IF (I-ITER) 65,75,65                                              MATIN066
   65 TEMP=A(I)                                                         MATIN067
      DO 70 J=2,N                                                       MATIN068
      IJM1=(J-2)*NDIM+I                                                 MATIN069
      IJ=(J-1)*NDIM+I                                                   MATIN070
      ITJM1=(J-2)*NDIM+ITER                                             MATIN071
      A(IJM1)=A(IJ)-A(ITJM1)*TEMP                                       MATIN072
   70 CONTINUE                                                          MATIN073
      IN=(N-1)*NDIM+I                                                   MATIN074
      ITN=(N-1)*NDIM+ITER                                               MATIN075
      A(IN)=-(TEMP*A(ITN))                                              MATIN076
   75 CONTINUE                                                          MATIN077
      DO 100 I=1,N                                                      MATIN078
      DO 80 J=I,N                                                       MATIN079
      IF (IROW(J)-I) 80,85,80                                           MATIN080
   80 CONTINUE                                                          MATIN081
   85 IF (I-J) 90,100,90                                                MATIN082
   90 DO 95 L=1,N                                                       MATIN083
      LI=(I-1)*NDIM+L                                                   MATIN084
      LJ=(J-1)*NDIM+L                                                   MATIN085
      TEMP=A(LI)                                                        MATIN086
      A(LI)=A(LJ)                                                       MATIN087
   95 A(LJ)=TEMP                                                        MATIN088
      IROW(J)=IROW(I)                                                   MATIN089
  100 CONTINUE                                                          MATIN090
      DO 125 I=1,N                                                      MATIN091
      DO 105 J=I,N                                                      MATIN092
      IF (ICOL(J)-I) 105,110,105                                        MATIN093
  105 CONTINUE                                                          MATIN094
  110 IF (I-J) 115,125,115                                              MATIN095
  115 DO 120 L=1,N                                                      MATIN096
      IL=(L-1)*NDIM+I                                                   MATIN097
      JL=(L-1)*NDIM+J                                                   MATIN098
      TEMP=A(IL)                                                        MATIN099
      A(IL)=A(JL)                                                       MATIN100
  120 A(JL)=TEMP                                                        MATIN101
      ICOL(J)=ICOL(I)                                                   MATIN102
  125 CONTINUE                                                          MATIN103
      IROW(NP1)=0                                                       MATIN104
      RETURN                                                            MATIN105
      END                                                               MATIN106
```

```
C    I N P U T  D A T A
TEST CASE 10C SPHERICAL TANK  3/4 FULL    BOND NUMBER = 10
   12    12  200     0 1.8        1.3         .000001   1.0        10.
0.0          0.0       .1216      .0047       .2438     .0194      .3665      .0464
.4874        .0895     .6008      .1535       .6960     .2429      .7319      .2978
.7564        .3594     .7657      .4274       .7527     .5024      .7209      .5576
.768         .510      .825       .436        .884      .339       .933       .233
.970         .108      .995       -.037       .999      -.181      .977       -.341
.925         -.510     .855       -.647
0.0          -1.131    .094       -1.126      .180      -1.113     .264       -1.093
.343         -1.070    .422       -1.036      .500      -0.997     .572       -0.949
.640         -0.898    .695       -0.845      .735      -0.805     .765       -0.774
0.0          -.056     0.0        -.116       0.0       -.181      0.0        -.273
0.0          -.369     0.0        -.484       0.0       -.581      0.0        -.702
0.0          -.821     0.0        -.976
1.0          -65.788   5.0
-.36708          3.14159267       1.0             0.0
    7
```

```
C       INPUT DATA
TEST CASE 11  SPHEROIDAL TANK 3/8 FULL    BOND NUMBER = 5
    9   12  200     0 1.8      1.3        .000001   1.0       5.
0.0       0.0       .164      .008       .325      .032      .486      .075
.633      .140      .768      .231       .827      .287      .877      .351
.917      .422      .942      .500       .951      .581      .939      .663
.964      .600      .980      .540       .9941     .454      1.00      .367
.994      .280      .980      .194       .954      .111
0.0       -.499     .151      -.489      .301      -.460     .447      -.407
.588      -.333     .718      -.235      .750      -.201     .800      -.154
.833      -.114     .865      -.070      .900      -.015     .923      .031
0.0       -.06      0.0       -.12       0.0       -.18      0.0       -.24
0.0       -.30      0.0       -.36       0.0       -.42
1.0       -16.12    5.0
-.394        1.35             1.0            0.0
    7
```

APPENDIX V.

LISTING AND INPUT SAMPLE OF SSHAPE

```
      PROGRAM SSHAPE (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     PROJECT  02-1846-02                                                SSH00200
C     A U G U S T    1 9 6 9                                             SSH00300
C     *LOW GRAVITY LIQUID-VAPOR INTERFACE SHAPES IN                      SSH00400
C      AXISYMMETRIC CONTAINERS *                                         SSH00500
C     C D C    6 6 0 0     F O R T R A N                                 SSH00600
      DIMENSION  Y(10),F(10)                                             SSH00700
      DIMENSION  XOR(2000),ZOR(2000),               YBX(200),YBZ(200)    SSH00800
      DIMENSION  TBDX(2),TBANG(2),ANS(3),ARG(3)                          SSH00900
      DIMENSION  XYB(200),XRB(200),XY(2000),XR(2000)                     SSH01000
      DIMENSION  XKF(3),ZKF(3),XKB(3),ZKB(3)                             SSH01100
      DIMENSION  S(2000),G(50),RS(50),SSV(50),NN(2),RX(50),Z(50)         SSH01200
       TYPE REAL  K0,K0SV,K0P,K2F,K2B                                    SSH01300
      CON1 = 1.74532925E-02                                              SSH01400
      CON2 = 57.2957795                                                  SSH01500
      PI = 3.14159265                                                    SSH01600
      NTAPEI = 5                                                         SSH01610
      NTAPEO = 6                                                         SSH01620
C*****R E A D    C O N T A I N E R    C O O R D I N A T E S              SSH01700
 2000 READ (NTAPEI,500) NP                                               SSH01800
  500 FORMAT ( 2I5 )                                                     SSH01900
      READ (NTAPEI,200) (YBX(I),I=1,NP)                                  SSH02200
  200 FORMAT ( 8F10.0 )                                                  SSH02300
      READ (NTAPEI,200) (YBZ(I),I=1,NP)                                  SSH02400
C*****C O N T A I N E R    V O L U M E                                   SSH02500
      NP1 = NP-1                                                         SSH02600
      VT = 0.                                                            SSH02700
      DO 95 I=1,NP1                                                      SSH02800
      VT = VT+.5*(YBX(I)**2+YBX(I+1)**2)*(YBZ(I)-YBZ(I+1))               SSH02900
   95 CONTINUE                                                           SSH03000
      VT = PI*VT                                                         SSH03100
   20 READ (NTAPEI,200) ALPHA,K0,Y0,BN,BETAD,DK0,DBC,DY0                 SSH03200
      READ (NTAPEI,200) THETA,DTHETA,RLGTH                               SSH03300
C*****N U M B E R    O F    E Q U A T I O N S                            SSH03400
      READ (NTAPEI,500) N,IBOPT                                          SSH03500
      READ (NTAPEI,500) NN(1),NN(2)                                      SSH03600
      DO 94 I=1,NP                                                       SSH03700
      YBX(I) = YBX(I)/RLGTH                                              SSH03800
      YBZ(I) = YBZ(I)/RLGTH                                              SSH03900
   94 CONTINUE                                                           SSH04000
      Y0 = Y0/RLGTH                                                      SSH04100
      DY0 = DY0/RLGTH                                                    SSH04200
      TPRINT = 2.                                                        SSH04300
      IPRINT =((TPRINT+DTHETA/10.)/DTHETA)                               SSH04400
      K0SV = K0                                                          SSH04500
      DK0SV = DK0                                                        SSH04600
      DK0 = 4.*DK0                                                       SSH04700
      YBX1 = YBX(1)                                                      SSH04800
      YBZ1 = YBZ(1)                                                      SSH04900
      II = 0                                                             SSH05000
      JJ = 0                                                             SSH05100
      JK = 0                                                             SSH05200
      TBDX(2) = K0                                                       SSH05300
 1000 NT = 0                                                             SSH05400
      Y(1) = Y0                                                          SSH05500
      Y(2) = 0.                                                          SSH05600
      X = THETA*CON1                                                     SSH05700
      H = DTHETA*CON1                                                    SSH05800
      WRITE (NTAPEO,300)                                                 SSH05900
  300 FORMAT (22H1 I N P U T    D A T A)                                 SSH06000
      WRITE (NTAPEO,305) ALPHA,K0,Y0,BN                                  SSH06100
  305 FORMAT (9H0ALPHA = ,E15.8,6H (DEG),4X,5HK0 = ,E15.8,6X,5HY0 = ,    SSH06200
```

```
   1 E15.8,4X,5HBN = ,E15.8)                                          SSH06300
     WRITE (NTAPEO,310)                                                SSH06400
 310 FORMAT (1H0,5X,5HTHETA,10X,3HXOR,12X,3HZOR,12X,4HY(1),12X,4HY(2)/)SSH06500
     XOR(1) = 0.                                                       SSH06600
     ZOR(1) = Y0                                                       SSH06700
     XDEG = 0,                                                         SSH06800
     WRITE(NTAPEO,315) XDEG,XOR(1),ZOR(1),Y(1),Y(2)                    SSH06900
     VU1 = 0,                                                          SSH07000
     ICCUNT = 1                                                        SSH07100
     K = 1                                                             SSH07200
     IJK = 2                                                           SSH07300
     DO 35 I=2,NP                                                      SSH07400
  75 F(1) = Y(2)                                                       SSH07500
     IF ( X,EQ,0, ) 125,130                                            SSH07600
 125 F(2) = Y0*(1,-K0)                                                 SSH07700
     GO TO 135                                                         SSH07800
 130 CONTINUE                                                          SSH07900
     TERM = Y(1)**2+Y(2)**2                                            SSH08000
     IF ( TERM-1,0E+60 ) 131,132,132                                   SSH08100
 132 XDEG = X*CON2                                                     SSH08200
     WRITE (NTAPEO,315) XDEG,Y(1),Y(2),F(1),F(2)                       SSH08300
     IF ( JJ,EQ,0 ) 777,778                                            SSH08400
 777 K0 = K0-DK0SV                                                     SSH08500
     GO TO 1000                                                        SSH08600
 778 CONTINUE                                                          SSH08700
     GO TO 74                                                          SSH08800
 131 CONTINUE                                                          SSH08900
     F(2) = ((2,*Y(1)**2+3,*Y(2)**2)/Y(1))-(Y(2)/Y(1)**2)*(1,/         SSH09000
    1          TAN(X))*(Y(1)**2+Y(2)**2)+(1,/Y(1))*(BN*(Y(1)*COS(X)-Y0)- SSH09100
    2         ((2,*K0)/Y0))*(SQRT(Y(1)**2+Y(2)**2))**3                 SSH09200
 135 CONTINUE                                                          SSH09300
     S = RKLDEQ ( N, Y, F, X, H, NT )                                  SSH09400
     IF ( S-1,0 ) 105,75,110                                           SSH09500
 105 STOP                                                              SSH09600
 110 TERM1 = SIN(X)                                                    SSH09700
     TERM2 = COS(X)                                                    SSH09800
     XDEG = X*CON2                                                     SSH09900
     TERM = YBZ(I-1)-YBZ(I)                                            SSH10000
     IF ( TERM,EQ,0,)465,470                                           SSH10100
 465 R = 1,0E+100                                                      SSH10200
     GO TO 475                                                         SSH10300
 470 R = (YBX(I)-YBX(I-1))/TERM                                        SSH10400
 475 CONTINUE                                                          SSH10500
     YX = (YBX(I-1)+R*YBZ(I-1))/(TERM1+R*TERM2)                        SSH10600
     YBX2 = YBX(I)                                                     SSH10700
     YBZ2 = YBZ(I)                                                     SSH10800
     YZ = SQRT(YBX(I)**2+YBZ(I)**2)                                    SSH10900
     K = K+1                                                           SSH11000
     XOR(K) = Y(1)*TERM1                                               SSH11100
     ZOR(K) = Y(1)*TERM2                                               SSH11200
     VU1 = VU1+,5*(XOR(K-1)**2+XOR(K)**2)*(ZOR(K-1)-ZOR(K))*PI         SSH11300
     IF ( YZ,EQ,0, ) 111,113                                           SSH11400
 111 THETAB = 90,*CON1                                                 SSH11500
     GO TO 112                                                         SSH11600
 113 THETAB = ASIN(YBX(I)/YZ)                                          SSH11700
 112 CONTINUE                                                          SSH11800
     IF ( Y(1)-YX ) 40,45,45                                           SSH11900
  40 CONTINUE                                                          SSH12000
     IF ( XDEG-90, ) 50,74,74                                          SSH12100
  50 CONTINUE                                                          SSH12200
     IF ( ICOUNT-IPRINT ) 115,120,115                                  SSH12300
 115 ICOUNT = ICOUNT+1                                                 SSH12400
```

```
      GO TO 117                                                         SSH12500
  120 CONTINUE                                                          SSH12600
      WRITE (NTAPEO,315) XDEG,XOR(K),ZOR(K),Y(1),Y(2)                   SSH12800
  315 FORMAT ( 8E15,7 )                                                 SSH12900
      ICCUNT = 1                                                        SSH13000
  117 IF ( X-THETAB ) 116,118,118                                       SSH13200
  116 YBX1 = YBX2                                                       SSH13300
      YBZ1 = YBZ2                                                       SSH13400
      GO TO 75                                                          SSH13500
  118 ISAVE = I                                                         SSH13600
      GO TO 35                                                          SSH13700
   45 GAMMA =(ATAN2(Y(2)*SIN(X)+Y(1)*COS(X),Y(1)*SIN(X)-Y(2)*COS(X)))*  SSH13800
     1 CCN2                                                             SSH13900
      IF ( ABS(GAMMA)-90, ) 46,46,74                                    SSH14000
   46 CONTINUE                                                          SSH14100
      ARG1 = -(YBX(I)-YBX(I-1))                                         SSH14200
      ARG2 = -(YBZ(I)-YBZ(I-1))                                         SSH14300
      IF ( ABS(ARG2),LE,(1,0E-08)) 47,48                                SSH14400
   47 PHI = 90,                                                         SSH14500
      GO TO 49                                                          SSH14600
   48 PHI = ATAN2(ARG1,ARG2)*CON2                                       SSH14700
   49 ANGLE = GAMMA+PHI                                                 SSH14800
      WRITE (NTAPEO,315) XDEG,XOR(K),ZOR(K),Y(1),Y(2)                   SSH14900
      WRITE (NTAPEO,320) GAMMA,PHI,ANGLE,Y0,K0                          SSH15000
  320 FORMAT (10H0 GAMMA = ,E15.8,8H  PHI = ,E15.8,10H  ALPHA = ,E15.8, SSH15100
     1 7H  Y0 = ,E15,8,7H  K0 = ,E15,8)                                 SSH15200
      IF ( ABS(ANGLE-ALPHA)-,5 ) 55,55,60                               SSH15300
   55 VU2 = 0,                                                          SSH15400
      JJ = 0                                                            SSH15500
      JK = 0                                                            SSH15600
      TBDX(2) = K0SV                                                    SSH15700
      YBXSV = YBX(ISAVE-1)                                              SSH15800
      YBZSV = YBZ(ISAVE-1)                                              SSH15900
      YBX(ISAVE-1) = XOR(K)                                             SSH16000
      YBZ(ISAVE-1) = ZOR(K)                                             SSH16100
      IJ = ISAVE-1                                                      SSH16200
      DO 100 J=IJ,NP1                                                   SSH16300
      VU2 = VU2+,5*(YBX(J)**2+YBX(J+1)**2)*(YBZ(J)-YBZ(J+1))*PI         SSH16400
  100 CONTINUE                                                          SSH16500
      YBX(ISAVE-1) = YBXSV                                              SSH16600
      YBZ(ISAVE-1) = YBZSV                                              SSH16700
      VU1 = VU1*RLGTH**3                                                SSH16800
      VU2 = VU2*RLGTH**3                                                SSH16900
      VU = VU1+VU2                                                      SSH17000
      VL = VT-VU                                                        SSH17100
      BETA  = VU/VT                                                     SSH17200
      WRITE (NTAPEO,325) VU1,VU2,VL,VL,VT,BETA                          SSH17300
  325 FORMAT (8H0 VU1 = ,E15,8,8H  VU2 = ,E15,8,7H  VU = ,E15,8,        SSH17400
     1 7H  VL = ,E15,8,7H  VT = ,E15,8//9H  BETA = ,E15,8)              SSH17500
      BB = BETAD-BETA                                                   SSH17600
      IF ( IBOPT,EQ,0 )  GO TO 145                                      SSH17700
      IF ( ABS(BB)-DBC ) 145,145,140                                    SSH17800
  140 CONTINUE                                                          SSH17900
      IF ( II ) 155,150,155                                             SSH18000
  150 II = II+1                                                         SSH18100
      ANS(1) = Y0                                                       SSH18200
      Y0 = Y0+DY0                                                       SSH18300
      ARG(1) = BETA                                                     SSH18400
      K0 = K0SV                                                         SSH18500
      DK0 = DK0SV                                                       SSH18600
      GO TO 1000                                                        SSH18700
  155 IF ( II-1 ) 165,160,165                                           SSH18800
```

```
160 II = II+1                                                          SSH18900
    ANS(2) = Y0                                                        SSH19000
    ARG(2) = BETA                                                      SSH19100
    Y0 = Y0+DY0                                                        SSH19200
    K0 = K0SV                                                          SSH19300
    DK0 = DK0SV                                                        SSH19400
    GO TO 1000                                                         SSH19500
165 IF ( II-2 ) 175,170,175                                            SSH19600
170 ANS(3) = Y0                                                        SSH19700
    II = II+1                                                          SSH19800
175 ARG(3) = BETA                                                      SSH19900
    IF ( ARG(1)-ARG(3) ) 405,400,400                                   SSH20000
400 AT = ARG(1)                                                        SSH20100
    ARG(1) = ARG(3)                                                    SSH20200
    ARG(3) = AT                                                        SSH20300
    TS = ANS(1)                                                        SSH20400
    ANS(1) = ANS(3)                                                    SSH20500
    ANS(3) = TS                                                        SSH20600
405 IF ( ARG(1)-ARG(2) ) 415,410,410                                   SSH20700
410 PP = ARG(1)                                                        SSH20800
    ARG(1) = ARG(2)                                                    SSH20900
    ARG(2) = PP                                                        SSH21000
    TS = ANS(1)                                                        SSH21100
    ANS(1) = ANS(2)                                                    SSH21200
    ANS(2) = TS                                                        SSH21300
415 IF ( ARG(2)-ARG(3) ) 425,420,420                                   SSH21400
420 ST = ARG(2)                                                        SSH21500
    ARG(2) = ARG(3)                                                    SSH21600
    ARG(3) = ST                                                        SSH21700
    TS = ANS(2)                                                        SSH21800
    ANS(2) = ANS(3)                                                    SSH21900
    ANS(3) = TS                                                        SSH22000
425 CONTINUE                                                           SSH22100
    IF ( BETAD-ARG(1) ) 435,430,430                                    SSH22200
430 IF ( BETAD-ARG(3) ) 440,445,445                                    SSH22300
440 IF ( BETAD-ARG(2) ) 450,455,455                                    SSH22400
450 ANSE = (ANS(1)*(ARG(2)-BETAD)-ANS(2)*(ARG(1)-BETAD))/(ARG(2)-      SSH22500
   1 ARG(1))                                                           SSH22600
    GO TO 460                                                          SSH22700
455 ANSE = (ANS(2)*(ARG(3)-BETAD)-ANS(3)*(ARG(2)-BETAD))/(ARG(3)-      SSH22800
   1 ARG(2))                                                           SSH22900
    GO TO 460                                                          SSH23000
435 ANSE = (ANS(1)*(ARG(2)-BETAD)+ANS(2)*(BETAD-ARG(1)))/(ARG(2)-      SSH23100
   1 ARG(1))                                                           SSH23200
    GO TO 460                                                          SSH23300
445 ANSE = (ANS(2)*(ARG(3)-BETAD)+ANS(3)*(BETAD-ARG(2)))/(ARG(3)-      SSH23400
   1 ARG(2))                                                           SSH23500
460 CONTINUE                                                           SSH23600
    WRITE (NTAPEO,700) ANSE                                            SSH23700
700 FORMAT (20H0 EXTRAPOLATED Y0 = ,E15.8)                             SSH23800
    Y0 = ANSE                                                          SSH23900
    ARG(1) = ARG(2)                                                    SSH24000
    ARG(2) = ARG(3)                                                    SSH24100
    ANS(1) = ANS(2)                                                    SSH24200
    ANS(2) = ANS(3)                                                    SSH24300
    ANS(3) = Y0                                                        SSH24400
    DK0 = DK0SV                                                        SSH24500
    K0 = K0SV                                                          SSH24600
    GO TO 1000                                                         SSH24700
145 CONTINUE                                                           SSH24800
    S(1) = 0.                                                          SSH24900
    SSV(1) = S(1)                                                      SSH25000
```

```
      NS = K-1
      DO 1  L=2,NS
      DELX = (XOR(L+1)-XOR(L))*RLGTH
      DELY = (ZOR(L)-ZOR(L+1))*RLGTH
      DELS = SQRT(DELX**2+DELY**2)
      S(L) = S(L-1)+DELS
    1 CONTINUE
      DO 5  JI=1,2
      IF (NN(JI).EQ.0 ) GO TO 6
      NN1 = NN(JI)-1
      NM = NN(JI)
      SI = S(NS)/NN1
      G(1) = 0.
      RS(1) = 1.
      DO 2  L=2,NN1
      DO 3  J=1,NS
      IF (S(J)-SI*(L-1)) 3,4,4
    4 SSV(L) = S(J)
      H1 = S(J-1)-S(J)
      H2 = S(J+1)-S(J)
      DEN = H1*H2*(H2-H1)
      AP0 = -(H2**2-H1**2)/DEN
      AP1 = H2**2/DEN
      AP2 = -(H1**2)/DEN
      APP0 = 2./(H1*H2)
      APP1 = (-2.*H2)/DEN
      APP2 = (2.*H1)/DEN
      RX(L) = XOR(J)*RLGTH
      Z (L) = (Y0-ZOR(J))*RLGTH
      RPS =(AP2*XOR(J+1)+AP0*XOR(J)+AP1*XOR(J-1))*RLGTH
      RPSS =(APP2*XOR(J+1)+APP0*XOR(J)+APP1*XOR(J-1))*RLGTH
      ZPS =(AP2*(Y0-ZOR(J+1))+AP0*(Y0-ZOR(J))+AP1*(Y0-ZOR(J-1)))*RLGTH
      ZPSS =(APP2*(Y0-ZOR(J+1))+APP0*(Y0-ZOR(J))+APP1*(Y0-ZOR(J-1)))*
     1 RLGTH
      CKAP2 = ZPS/(XOR(J)*RLGTH)
      CKAP1 = RPS*ZPSS-ZPS*RPSS
      G(L) = CKAP2**2+CKAP1**2
      RS(L) = RPS
      GO TO 2
    3 CONTINUE
    2 CONTINUE
      SSV(NM) = S(NS)
      G(NM) = G(NN1)
      RS(NM) = RS(NN1)
      RX(NM) = XOR(K)*RLGTH
      Z(NM) = (Y0-ZOR(K))*RLGTH
      PRINT 314
  314 FORMAT(1H1,14X,*G*,20X,*RS*,19X,*S*,19X,*XR*,19X,*XY*/)
      WRITE (NTAPE0,319) (I,G(I),RS(I),SSV(I),RX(I),Z(I),I=1,NM)
  319 FORMAT ( I5,5E20.8 )
    5 CONTINUE
    6 CONTINUE
      YBX(ISAVE-1) = YBXSV
      YBZ(ISAVE-1) = YBZSV
      DO 605 J=1,NP
      XRB(J) = YBX(J)*RLGTH
      XYB(J) =(Y0-YBZ(J))*RLGTH
  605 CONTINUE
      XKF(1) = XOR(K-2)
      XKF(2) = XOR(K-1)
      XKF(3) = XOR(K)
      ZKF(1) =-ZOR(K-2)+ZOR(1)
```

```
      ZKF(2) =-ZOR(K-1)+ZOR(1)
      ZKF(3) =-ZOR(K)+ZOR(1)
      CALL SFIT (YR,XK3,XK,YR2,XKF,ZKF)
       K2F = ABS(XK )
      FRR2 =  K2F*(1.+YR2**2)**(1.5)
      XKB(1) = YBX(I-1)
      XKB(2) = XOR(K)
      XKB(3) = YBX(I)
      ZKB(1) =-YBZ(I-1)+ZOR(1)
      ZKB(2) =-ZOR(K)+ZOR(1)
      ZKB(3) =-YBZ(I)+ZOR(1)
      CALL SFIT (YR,XK3,XK,YR2,XKB,ZKB)
       K2B = ABS(XK )
      ZRR2 =  K2B*(1.+YR2**2)**(1.5)
      CGAM = (1./SIN(ANGLE*CON1))*( K2B-COS(ANGLE*CON1)* K2F)
      WRITE (NTAPEO,805) K2F,FRR2,K2B,ZRR2,CGAM
  805 FORMAT (9H0  K2F = ,E15.8,9H  FRR2 = ,E15.8//7H K2B = ,E15.8,
     1 9H  ZRR2 = ,E15.8//8H CGAM = ,E15.8)
      DO 606 J=1,K
      XR( J)= XOR(J)*RLGTH
      XY( J)=(Y0-ZOR(J))*RLGTH
  606 CONTINUE
      Y0 = Y0*RLGTH
      WRITE (NTAPEO,318) Y0
  318 FORMAT (7H0 Y0 = ,E15.8)
      WRITE (NTAPEO,335)
  335 FORMAT (1H1,9X,3HXRB,17X,3HXYB/)
      WRITE (NTAPEO,317) (XRB(J),XYB(J),J=1,NP)
  317 FORMAT ( 2E20.8 )
      SUM1 = 0.
      DO 610 J=2,K
      SUM1 = SUM1+.5*(XR(J)**2*XY(J)+XR(J-1)**2*XY(J-1))*(XY(J)-XY(J-1))
  610 CONTINUE
      SUM2 = 0.
      IJ = ISAVE
      XRBSV = XRB(IJ)
      XYBSV = XYB(IJ)
      XRB(IJ) = XOR(K)*RLGTH
      XYB(IJ) = (Y0-ZOR(K))*RLGTH
      DO 615 J=IJ,NP1
      SUM2 = SUM2+.5*(XRB(J)**2*XYB(J)+XRB(J+1)**2*XYB(J+1))*(XYB(J+1)-
     1 XYB(J))
  615 CONTINUE
      BZCGU = (PI*(SUM1+SUM2))/VT
      XYB(IJ) = XYBSV
      XRB(IJ) = XRBSV
      SUM1 = 0.
      DO 620 J=1,NP1
      SUM1 = SUM1+.5*(XRB(J)**2*XYB(J)+XRB(J+1)**2*XYB(J+1))*(XYB(J+1)-
     1 XYB(J))*PI
  620 CONTINUE
      ZCGT = SUM1/VT
      ZCG = (ZCGT-BZCGU)/(1.-BETA)
      WRITE (NTAPEO,800) ZCGT,BZCGU,ZCG
  800 FORMAT (9H0 ZCGT = ,E15.8,10H  BZCGU = ,E15.8,8H  ZCG = ,E15.8)
      GO TO 2000
   60 CONTINUE
      IF ( ANGLE-ALPHA ) 74,65,71
   71 IF ( JJ-1 ) 72,73,72
   72 TBDX(1) = K0
      TBANG(1) = ANGLE
      JJ = JJ+1
```

```
      GO TO 70                                                          SSH37500
   73 TBDX(2) = K0                                                      SSH37600
      TBANG(2) = ANGLE                                                  SSH37700
      IF ( JK ) 76,76,77                                                SSH37800
   77 K0 = .5*(K0P+TBDX(2))                                             SSH37900
      GO TO 1000                                                        SSH38000
   76 K0 = (TBDX(1)*(TBANG(2)-ALPHA)-TBDX(2)*(TBANG(1)-ALPHA))/         SSH38100
     1 (TBANG(2)-TBANG(1))                                              SSH38200
      JJ = 1                                                            SSH38300
      TBDX(1) = TBDX(2)                                                 SSH38400
      TBANG(1) = TBANG(2)                                               SSH38500
      GO TO 1000                                                        SSH38600
   65 DK0 = DK0/4.                                                      SSH38700
      K0 = K0 - DK0                                                     SSH38800
      GO TO 1000                                                        SSH38900
   70 DK0 = DK0/4.                                                      SSH39000
      K0 = K0 + DK0                                                     SSH39100
      GO TO 1000                                                        SSH39200
   74 K0P = K0                                                          SSH39300
      K0 = .5*(K0+TBDX(2))                                              SSH39400
      JK = 1                                                            SSH39500
      GO TO 1000                                                        SSH39600
   35 CONTINUE                                                          SSH39700
      END                                                               SSH39800
```

```
      FUNCTION RKLDEQ(N,Y,F,X,H,NT)                                     RKLDQ  1
C D2 UCSD RKLDEQ RUNGE-KUTTA-GILL LINEAR DIFFERENTIAL EQUATION SOLVER   F-63   X
C     D2 UCSD RKLDEQ                                                       F 63
C     MODIFIED MAY 1963 (Q REMOVED FROM CALLING SEQUENCE)
C     TEST OF ALGOL ALGORITHM
      DIMENSION Y(10),F(10),G(10)                                       RKLDQ  2
C     REAL X,H--INTEGER N,NT--COMMENT--BEGIN INTEGER I,J,L-REAL A
      NT=NT+1                                                           RKLDQ  3
      GO TO (1,2,3,4),NT                                                RKLDQ  4
C     GO TO S(NT)
 1    DO 11 J=1,N                                                       RKLDQ  5
 11   Q(J)=0.                                                           RKLDQ  6
      A=.5                                                              RKLDQ  7
      X=X+H/2.                                                          RKLDQ  8
      GO TO 5                                                           RKLDQ  9
 2    A=.29289321881                                                    RKLDQ 10
      GO TO 5                                                           RKLDQ 11
 3    A=1.7071067812                                                    RKLDQ 12
      X=X+H/2.                                                          RKLDQ 13
      GO TO 5                                                           RKLDQ 14
 4    DO 41 I=1,N                                                       RKLDQ 15
 41   Y(I)=Y(I)+H*F(I)/6.-Q(I)/3.                                       RKLDQ 16
      NT=0                                                              RKLDQ 17
      RKLDEQ=2.                                                         RKLDQ 18
      GO TO 6                                                           RKLDQ 19
 5    DO 51 L=1,N                                                       RKLDQ 20
      Y(L)=Y(L)+A*(H*F(L)-Q(L))                                         RKLDQ 21
 51   Q(L)=2.*A*H*F(L)+(1.-3.*A)*Q(L)                                   RKLDQ 22
      RKLDEQ=1.                                                         RKLDQ 23
 6    CONTINUE                                                          RKLDQ 24
      RETURN                                                            RKLDQ 25
      END                                                               RKLDQ 26
```

```
      SUBROUTINE SFIT(YR,XK3,XK,YR2,R,YT)
      FIT QUADRATIC THROUGH THREE POINTS
      DIMENSION  R(3),YT(3)
      IF(ABS((YT(3)-YT(2))/(R(3)-R(2))).GT.1.0) GO TO 30
      IF(ABS((YT(2)-YT(1))/(R(2)-R(1))).GT.1.0) GO TO 30
      XTEMP=(R(1)**2-R(3)**2)*(R(2)-R(3))-(R(2)**2-R(3)**2)*(R(1)-R(3))
      YTEMP=(YT(1)-YT(3))*(R(2)-R(3))-(YT(2)-YT(3))*(R(1)-R(3))
      AD=YTEMP/XTEMP
      AB=(YT(2)-YT(3)-AD*(R(2)**2-R(3)**2))/(R(2)-R(3))
      YR=2.0*AD*R(3)+AB
      YR2=2.0*AD*R(2)+AB
      YRR=2.0*AD
      XK=YRR/(1.0+YR2**2)**1.5
      H2=R(2)-R(3)
      H1=R(1)-R(3)
      F2=YT(2)
      F1=YT(1)
      F0=YT(3)
      YRR=2.*(H2*(F1-F0)-H1*(F2-F0))/(H1*H2*(H1-H2))
      XK3=YRR/(1.0+YR**2)**1.5
      RETURN
30    YTEMP=(YT(1)**2-YT(3)**2)*(YT(2)-YT(3))-(YT(2)**2-YT(3)**2)*
     1      (YT(1)-YT(3))
      XTEMP=(R(1)-R(3))*(YT(2)-YT(3))-(R(2)-R(3))*(YT(1)-YT(3))
      AD=XTEMP/YTEMP
      AB=(R(2)-R(3)-AD*(YT(2)**2-YT(3)**2))/(YT(2)-YT(3))
      YR=1.0/(2.0*AD*YT(3)+AB)
      YR2=1.0/(2.0*AD*YT(2)+AB)
      YRR=-2.0*AD*YR2**3
      XK=YRR/(1.0+YR2**2)**1.5
      H2=YT(2)-YT(3)
      H1=YT(1)-YT(3)
      F2=R(2)
      F1=R(1)
      F0=R(3)
      RYY=2.*(H2*(F1-F0)-H1*(F2-F0))/(H1*H2*(H1-H2))
      YRR3=-RYY*YR**3
      XK3=YRR3/(1.0+YR**2)**1.5
      RETURN
      END
```

```
C      I N P U T  D A T A
C      SPHEROIDAL TANK
   37
0.          0.0756     0.1509     0.2262     0.3010     0.3740     0.447      0.519
0.588       0.654      0.718      0.778      0.833      0.861      0.923      0.954
0.980       0.994      1.000      0.994      0.980      0.954      0.923      0.861
0.833       0.778      0.718      0.654      0.588      0.519      0.447      0.374
0.301       0.2262     0.1509     0.0756     0.
1.732       1.730      1.722      1.710      1.693      1.668      1.640      1.607
1.566       1.520      1.468      1.411      1.347      1.277      1.202      1.122
1.039       0.953      0.866      0.779      0.693      0.610      0.530      0.455
0.385       0.321      0.264      0.212      0.166      0.125      0.092      0.064
0.039       0.022      0.010      0.002      0.
5.          0.         0.50       5.         0.625      0.2        0.005      0.50
0.          0.05       1.0
    2    1
   17   33
```